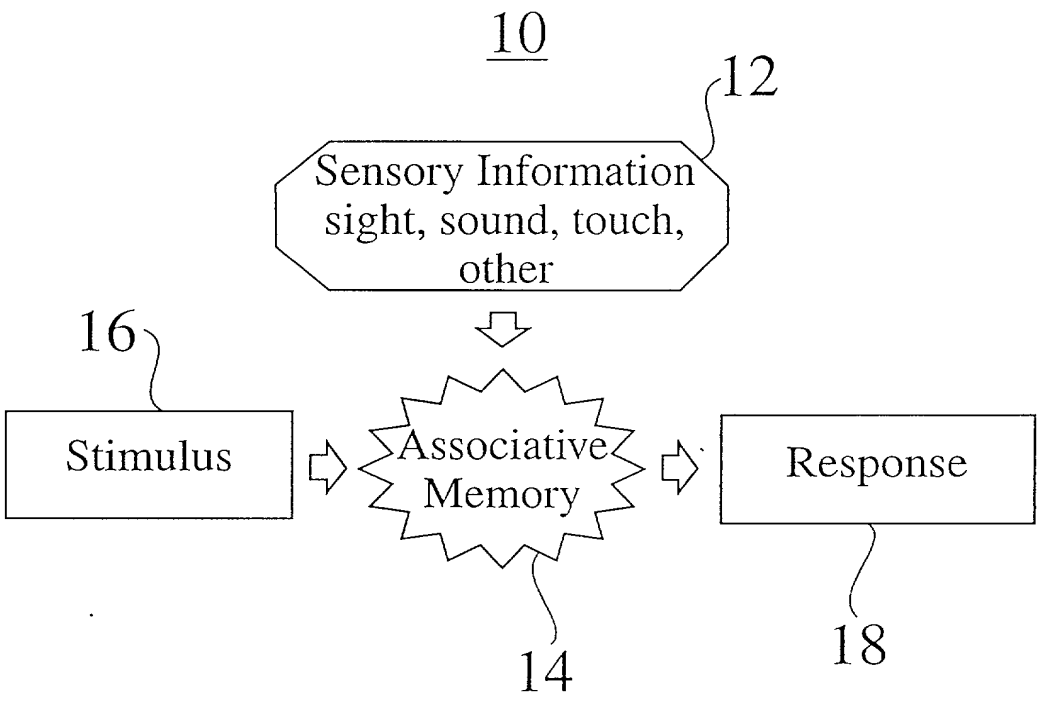

10
12
Sensory Information
sight, sound, touch,
other
16
Stimulus
Associative
Memory
Response
14
18

Fig. 1

Patent Application

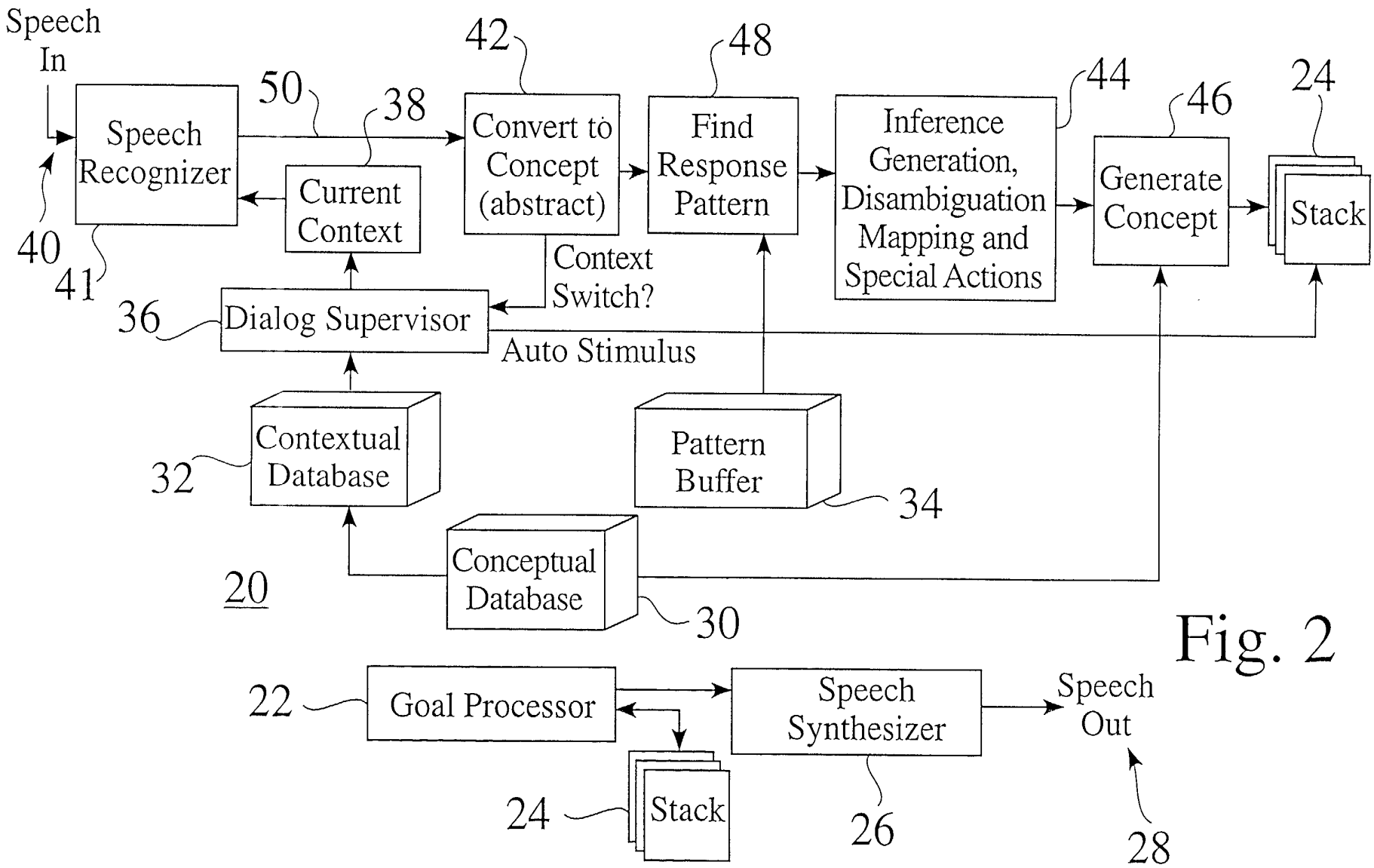


Fig. 2

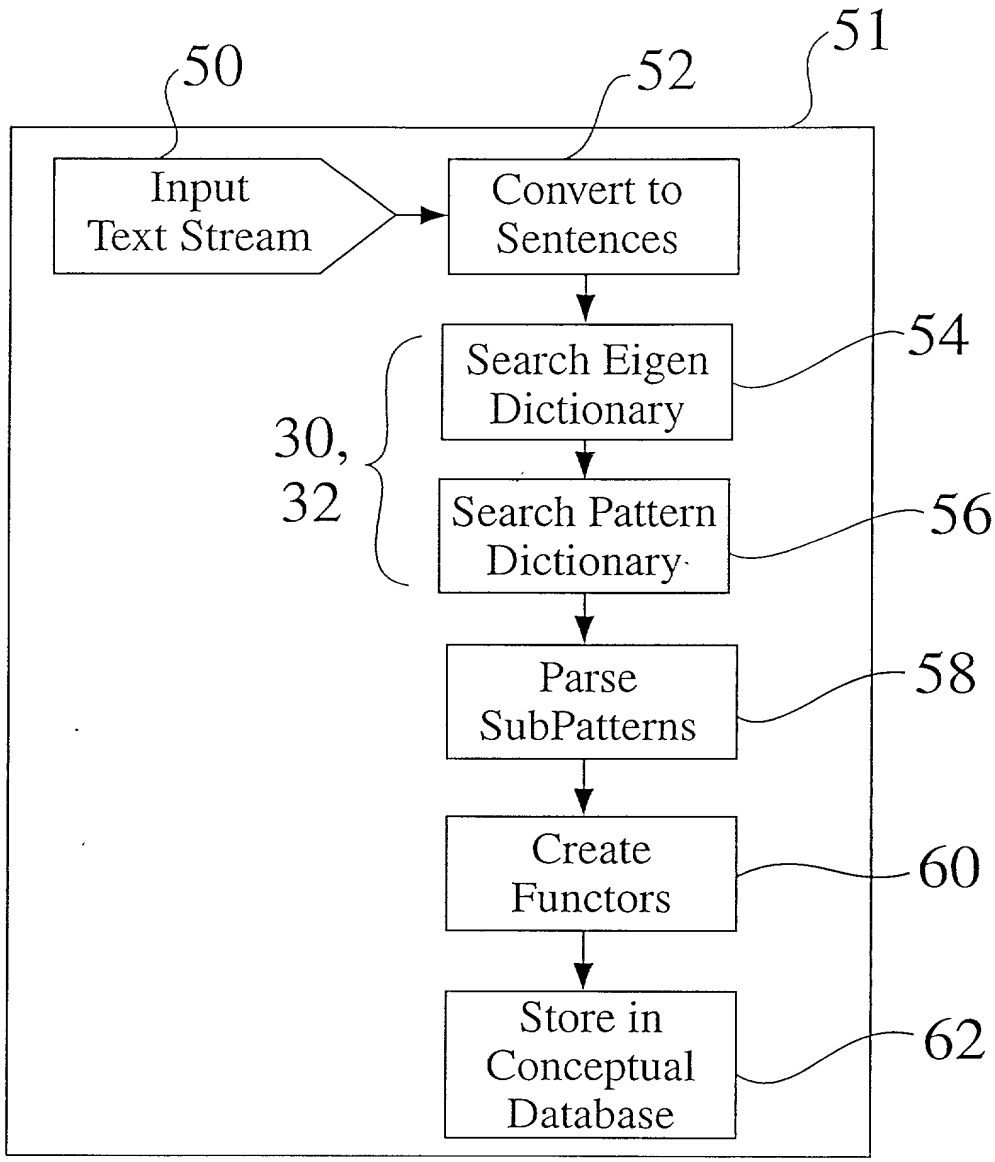

50
52
51
Input Text Stream
Convert to Sentences
Search Eigen Dictionary
54
30, 32
Search Pattern Dictionary
56
Parse SubPatterns
58
Create Functors
60
Store in Conceptual Database
62

Fig. 3

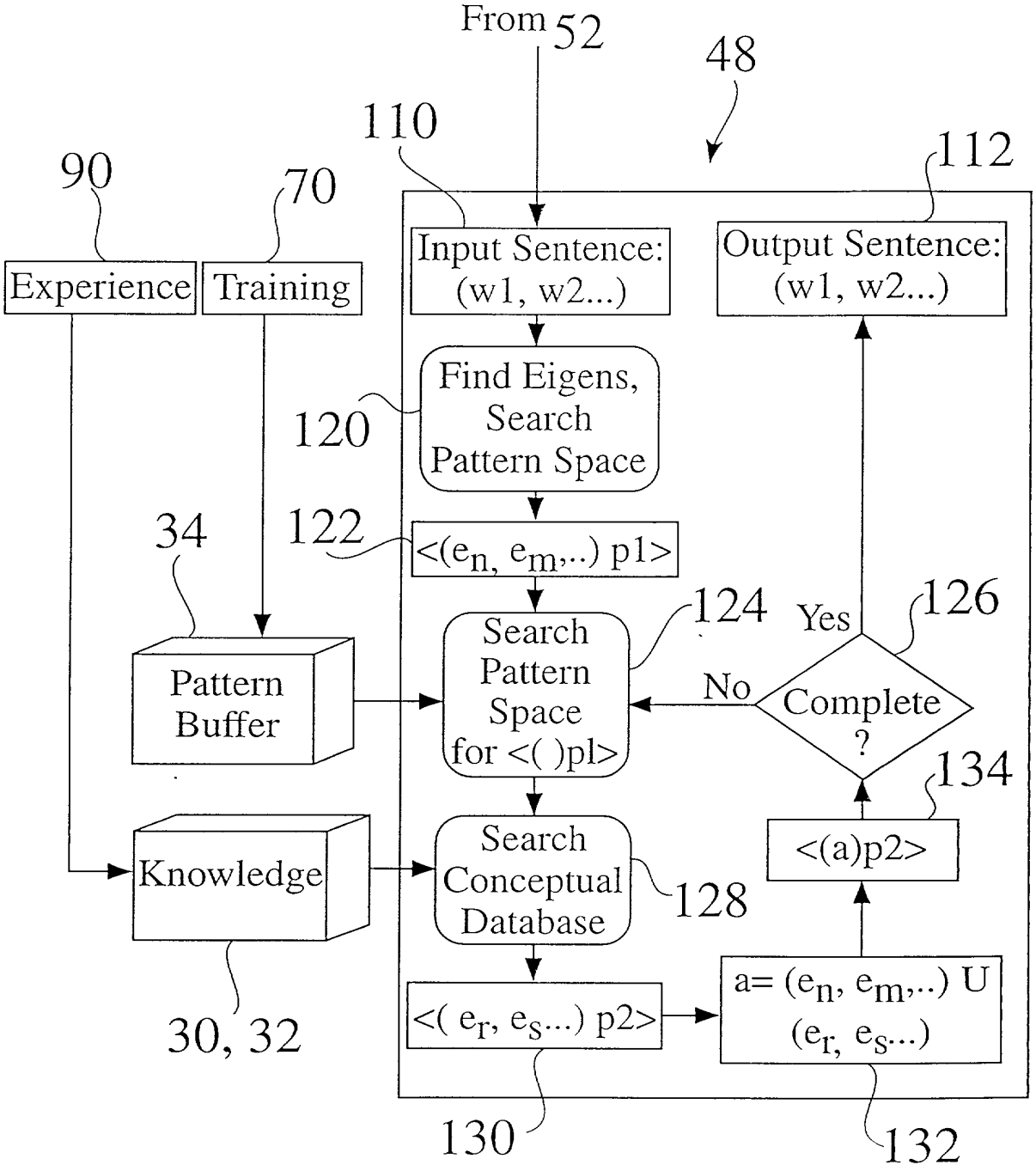

From 52
48
110
112
90
70
Experience
Training
Input Sentence: (w1, w2...)
Output Sentence: (w1, w2...)
Find Eigens, Search Pattern Space
120
34
122
<(en, em,..) p1>
Pattern Buffer
Search Pattern Space for <( )pl>
124
Yes
126
No
Complete ?
134
<(a)p2>
Knowledge
Search Conceptual Database
128
<(er, es...) p2>
a= (en, em,..) U (er, es...)
30, 32
130
132

Fig. 3A

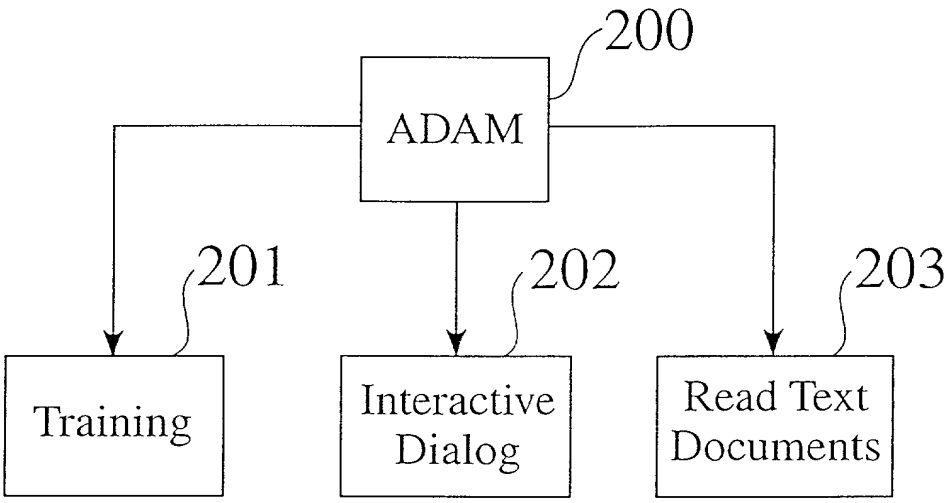


Fig. 4

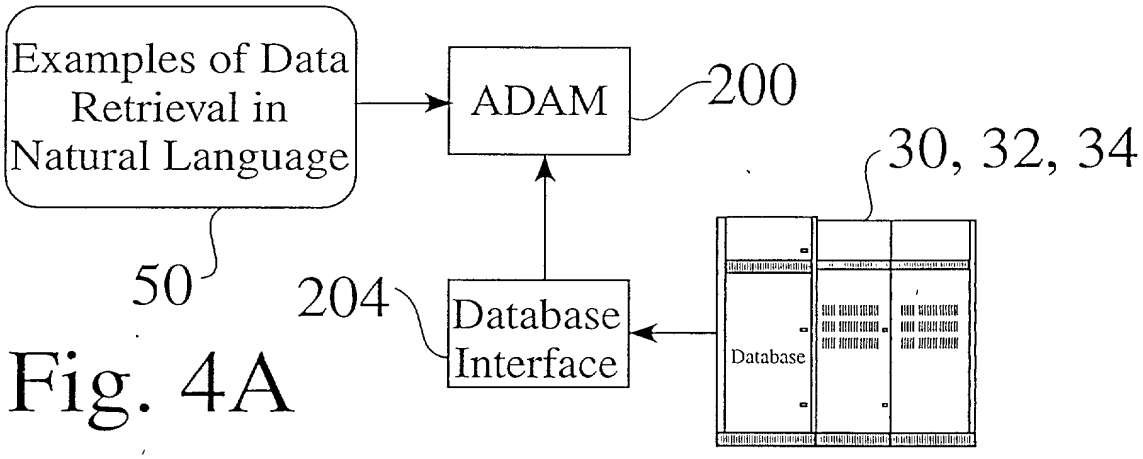


Fig. 4A

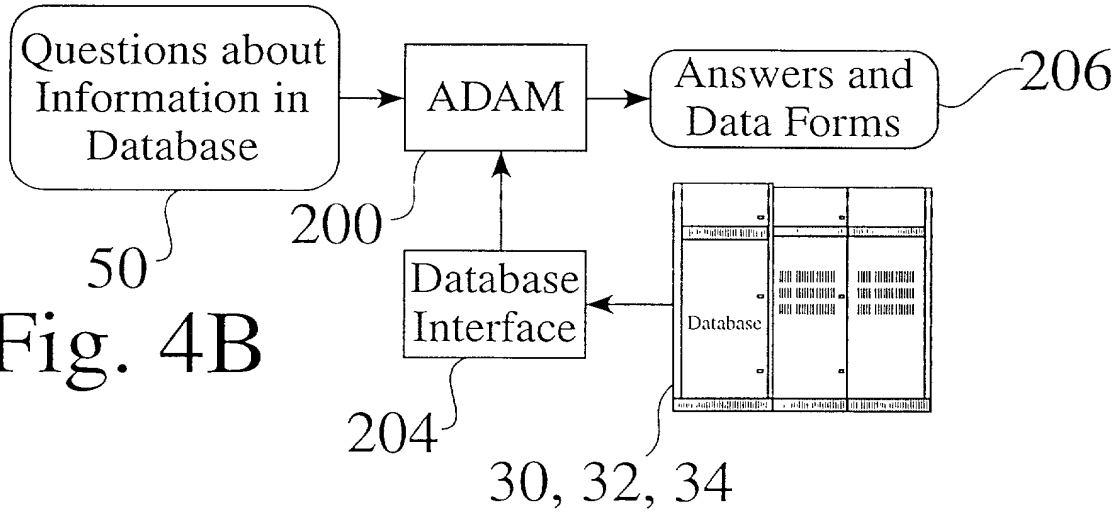


Fig. 4B

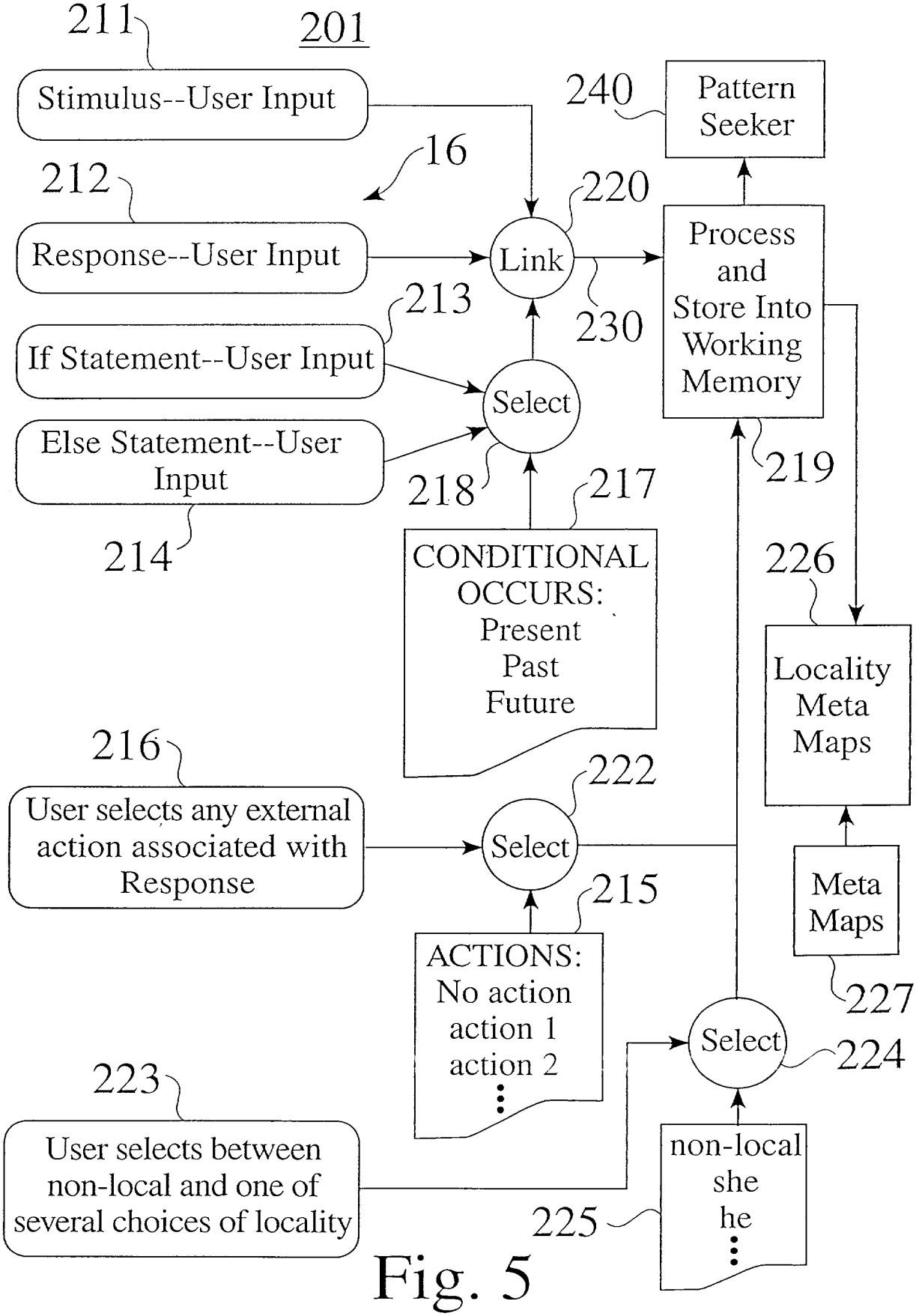

211
201
Stimulus--User Input
240
Pattern Seeker
16
212
220
Response--User Input
Link
Process and Store Into Working Memory
213
230
If Statement--User Input
Select
Else Statement--User Input
219
218
217
214
CONDITIONAL OCCURS:
Present
Past
Future
226
Locality Meta Maps
216
222
User selects any external action associated with Response
Select
Meta Maps
215
ACTIONS:
No action
action 1
action 2
227
Select
224
223
User selects between non-local and one of several choices of locality
non-local
she
he
225

Fig. 5

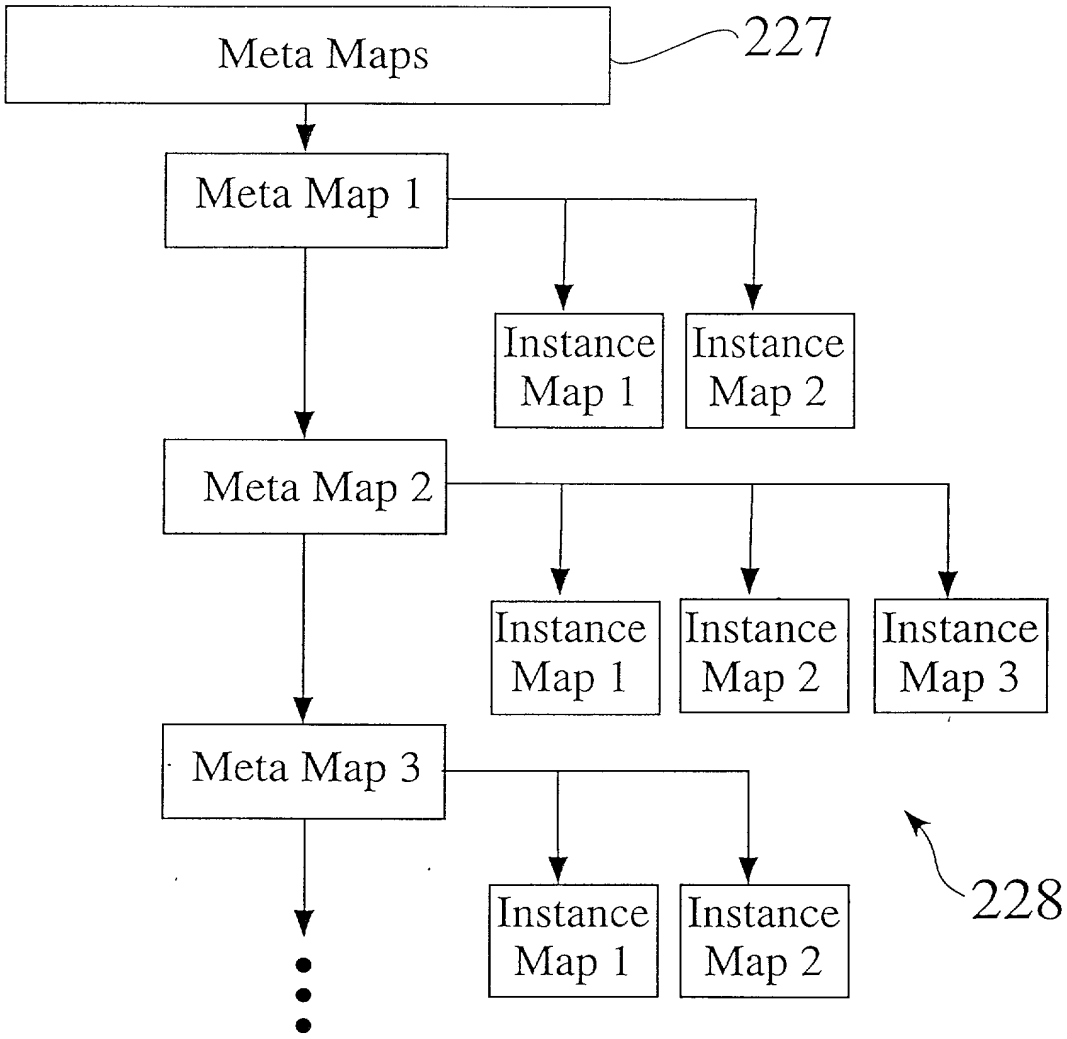

Meta Maps
227
Meta Map 1
Instance Map 1
Instance Map 2
Meta Map 2
Instance Map 1
Instance Map 2
Instance Map 3
Meta Map 3
Instance Map 1
Instance Map 2
228

Fig. 5A

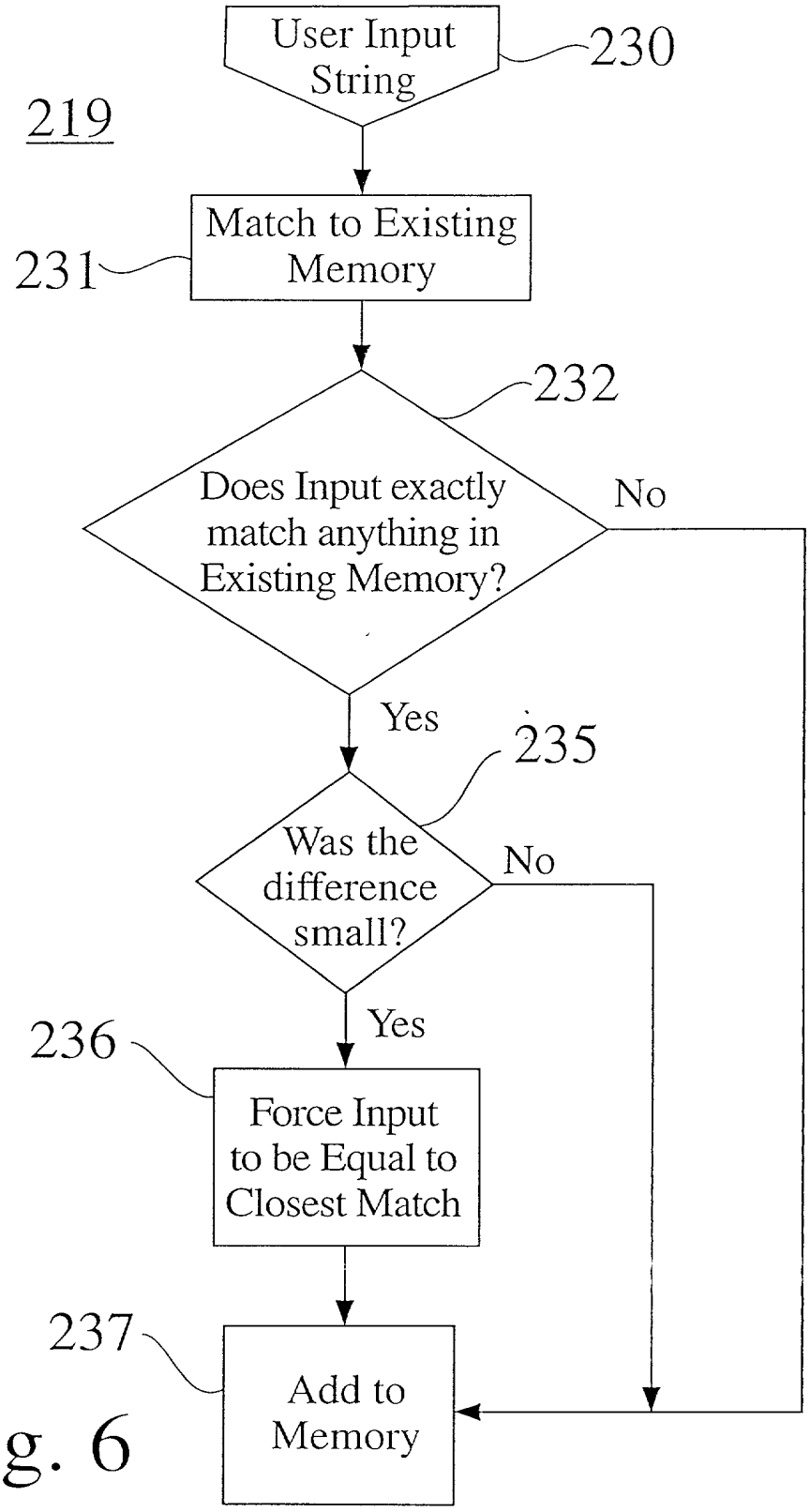

User Input String
230
219
Match to Existing Memory
231
232
Does Input exactly match anything in Existing Memory?
No
Yes
235
Was the difference small?
No
Yes
236
Force Input to be Equal to Closest Match
237
Add to Memory

Fig. 6

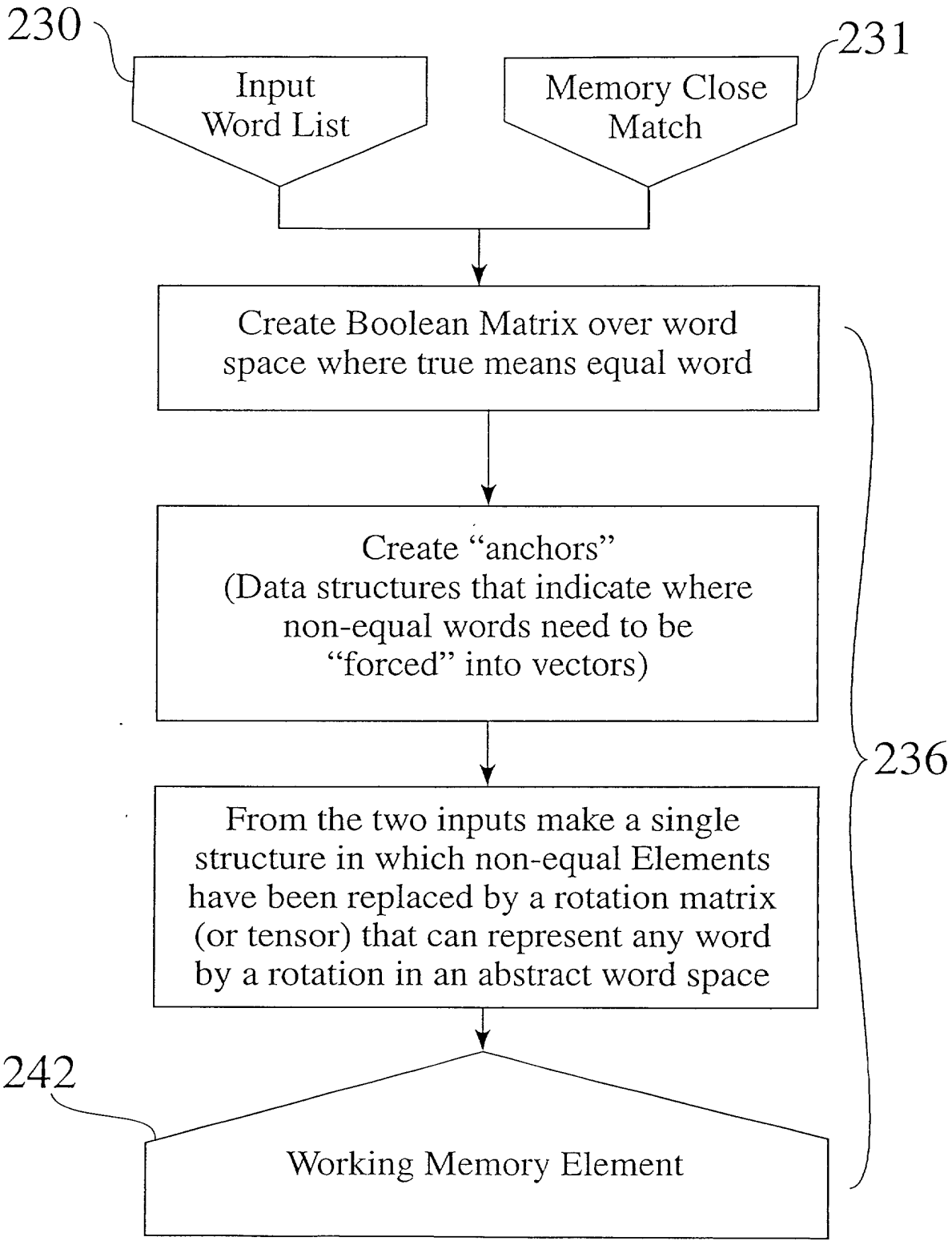

230
Input
Word List
231
Memory Close
Match
Create Boolean Matrix over word space where true means equal word
Create "anchors"
(Data structures that indicate where non-equal words need to be "forced" into vectors)
From the two inputs make a single structure in which non-equal Elements have been replaced by a rotation matrix (or tensor) that can represent any word by a rotation in an abstract word space
236
242
Working Memory Element

Fig. 7

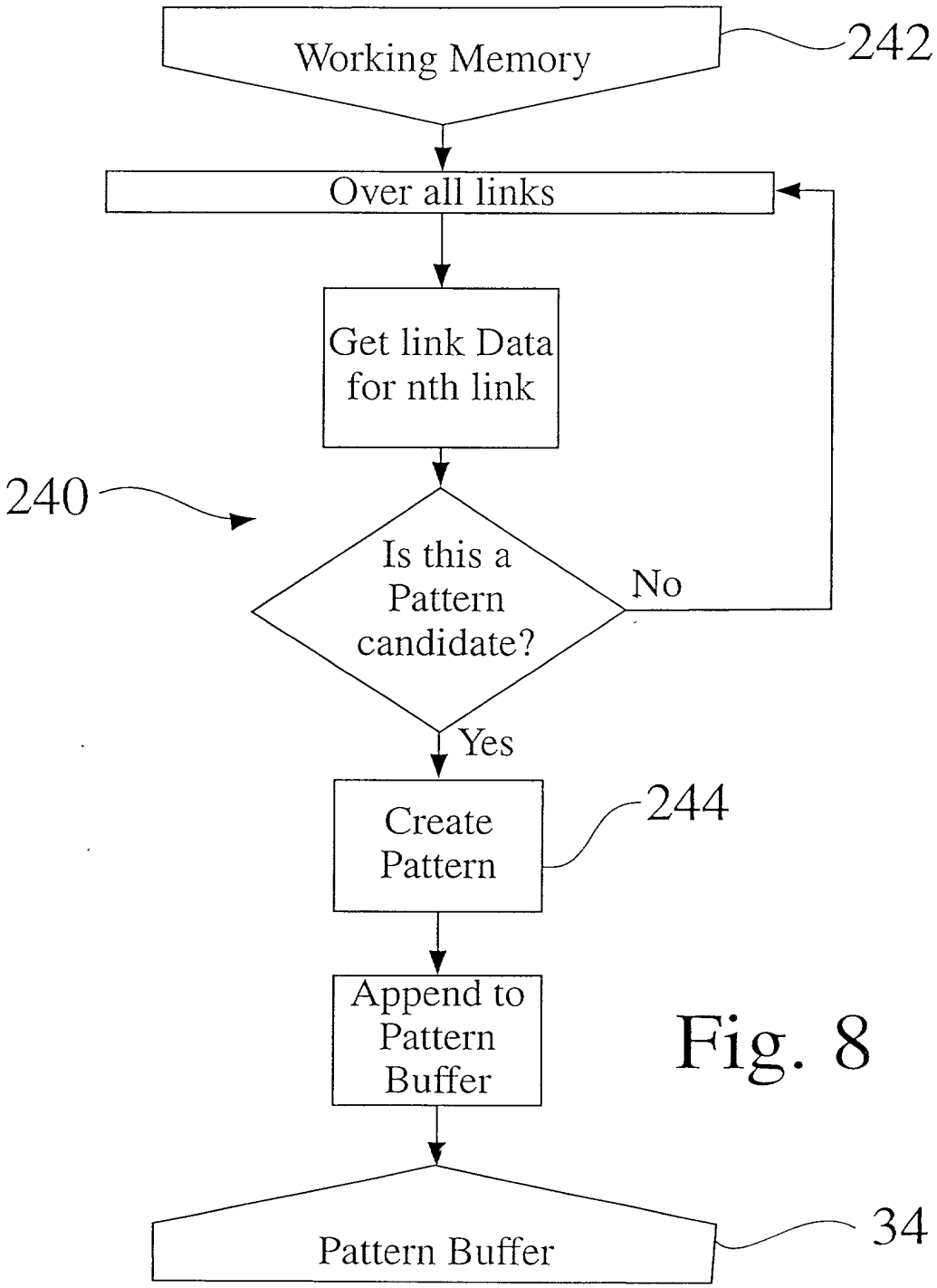

Working Memory
242
Over all links
Get link Data for nth link
240
Is this a Pattern candidate?
No
Yes
Create Pattern
244
Append to Pattern Buffer
Pattern Buffer
34

Fig. 8

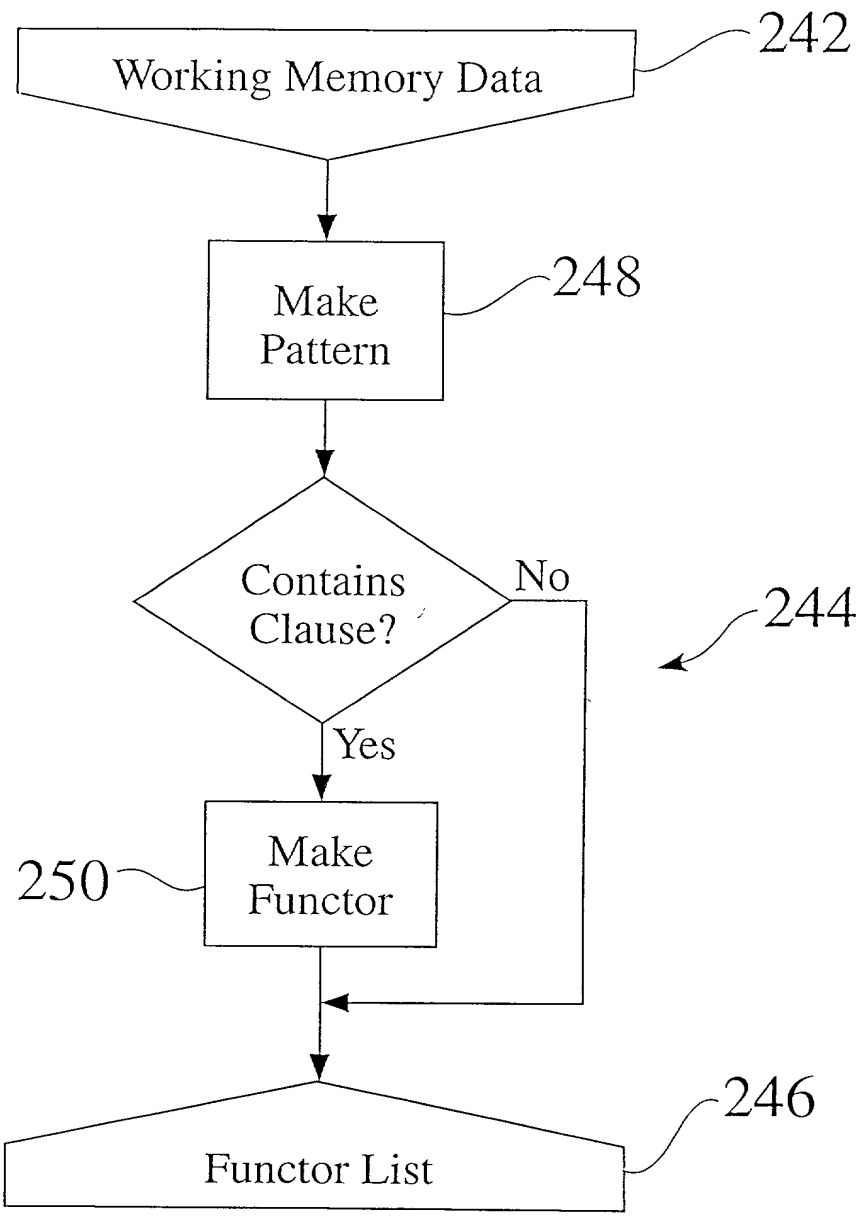

Working Memory Data
242
Make Pattern
248
Contains Clause?
No
244
Yes
Make Functor
250
Functor List
246

Fig. 9

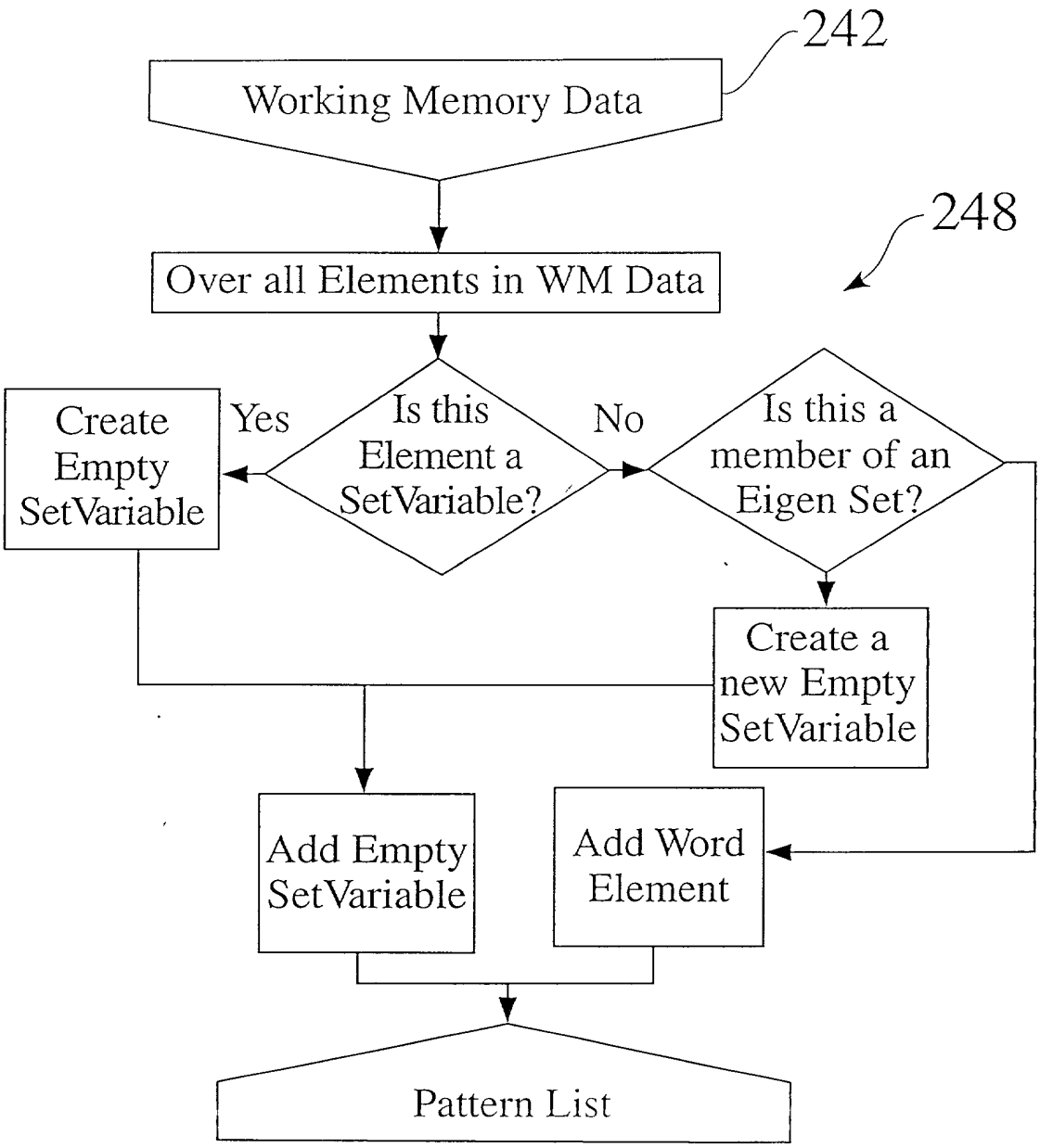

242
Working Memory Data
248
Over all Elements in WM Data
Is this Element a SetVariable?
Yes
Create Empty SetVariable
No
Is this a member of an Eigen Set?
Create a new Empty SetVariable
Add Empty SetVariable
Add Word Element
Pattern List

Fig. 10

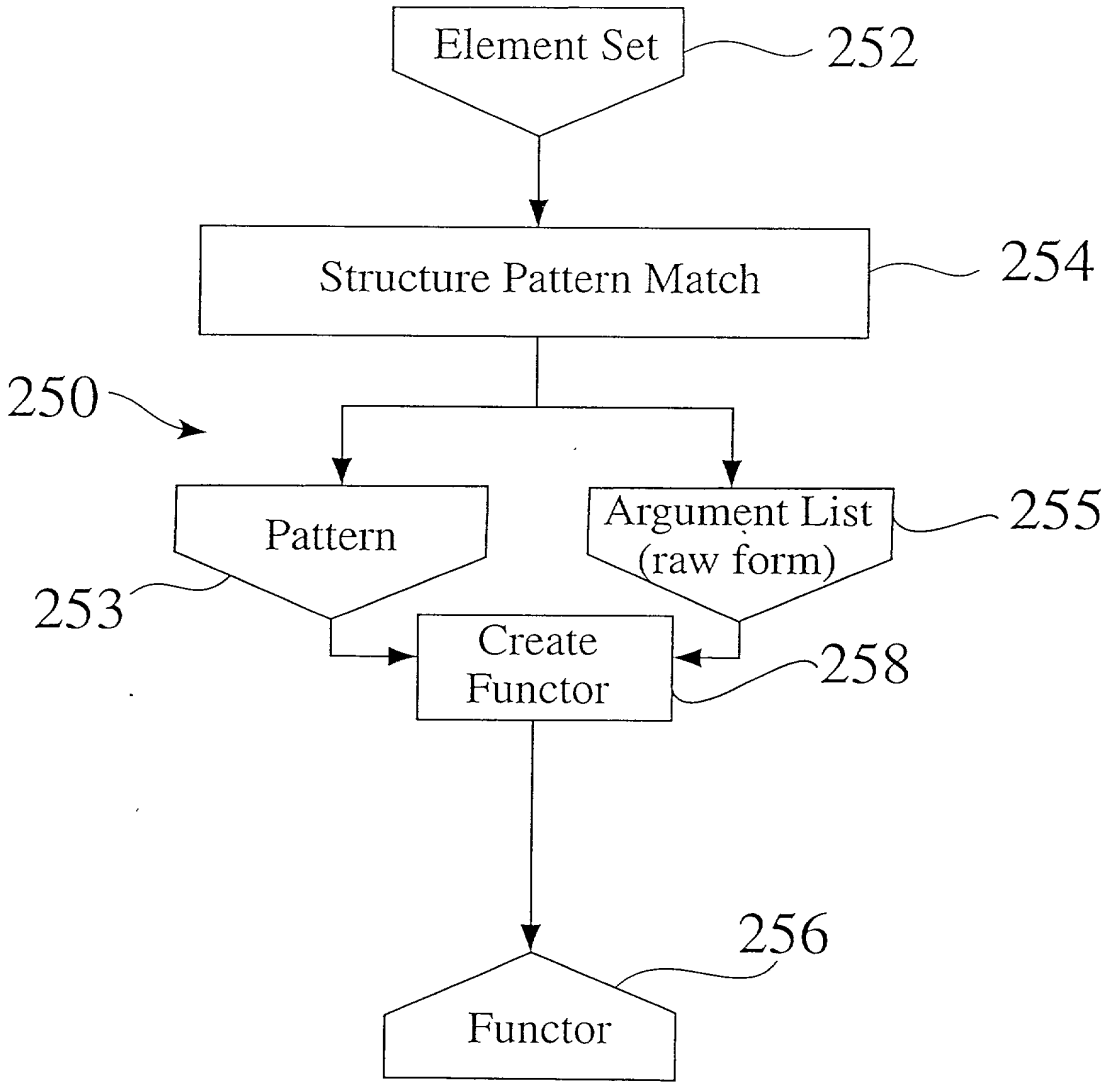

Element Set
252
Structure Pattern Match
254
250
Pattern
253
Argument List
(raw form)
255
Create
Functor
258
256
Functor

Fig. 11

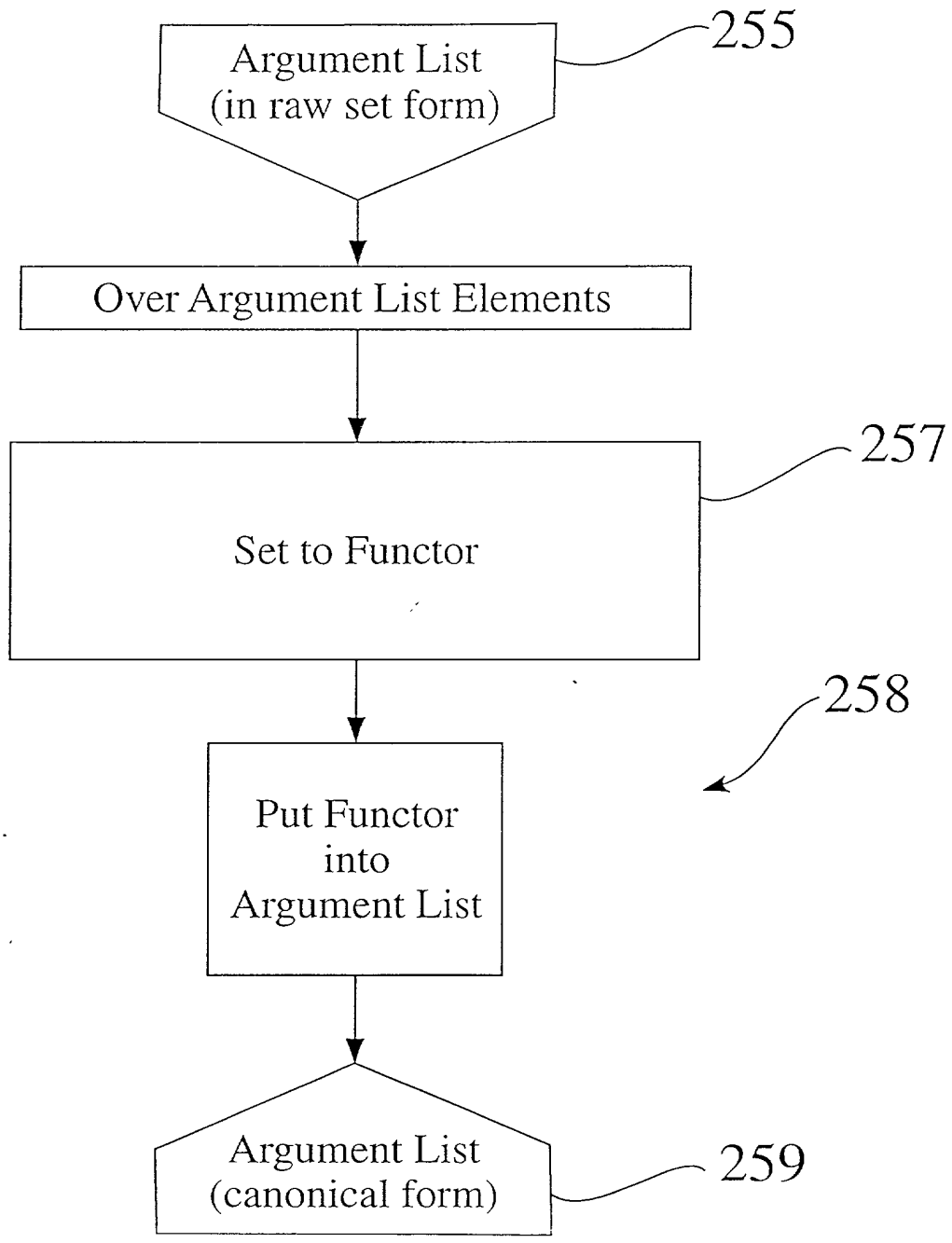

Argument List
(in raw set form)
255
Over Argument List Elements
Set to Functor
257
258
Put Functor
into
Argument List
Argument List
(canonical form)
259

Fig. 11A

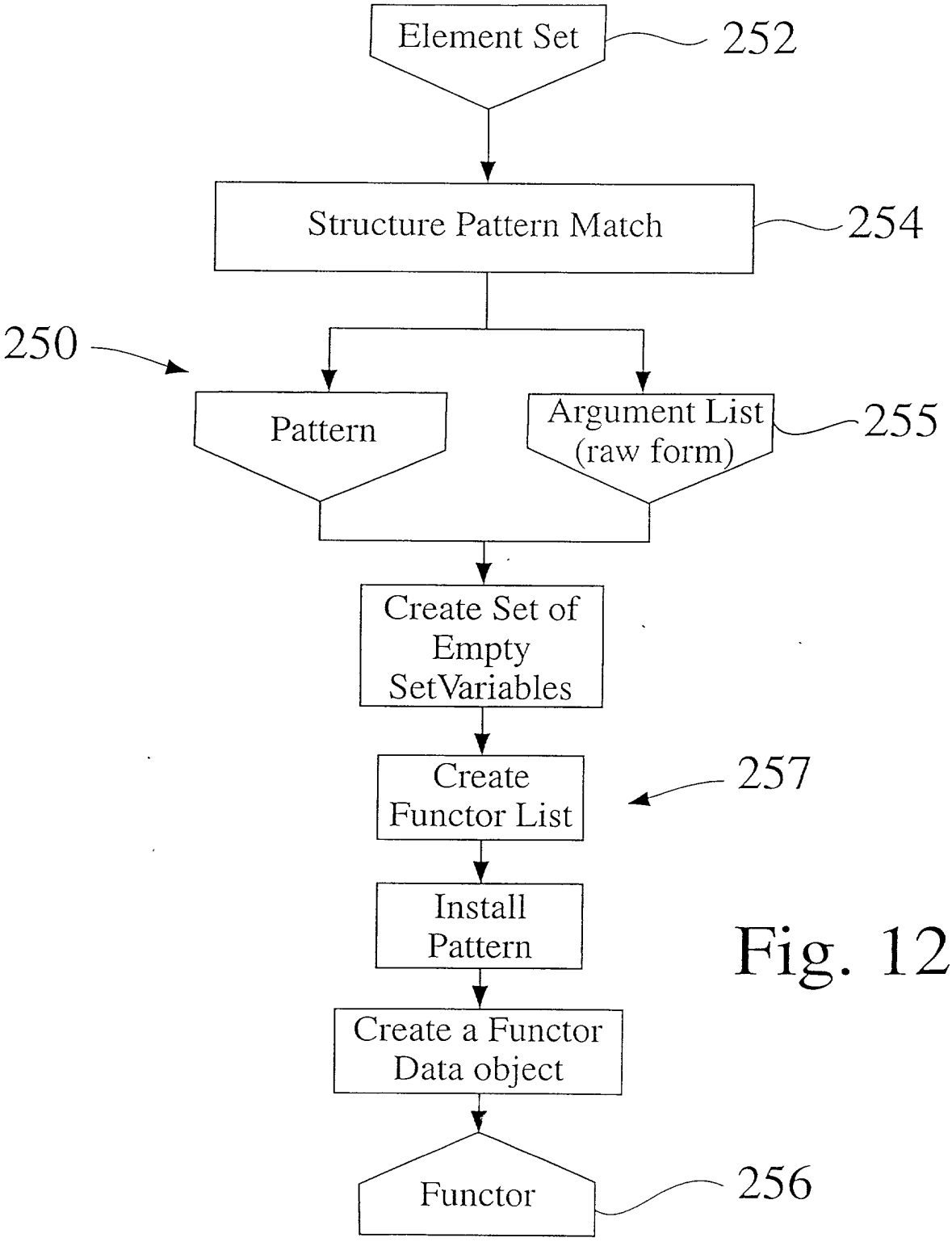

Element Set
252
Structure Pattern Match
254
250
Pattern
Argument List
(raw form)
255
Create Set of
Empty
SetVariables
Create
Functor List
257
Install
Pattern
Create a Functor
Data object
Functor
256

Fig. 12

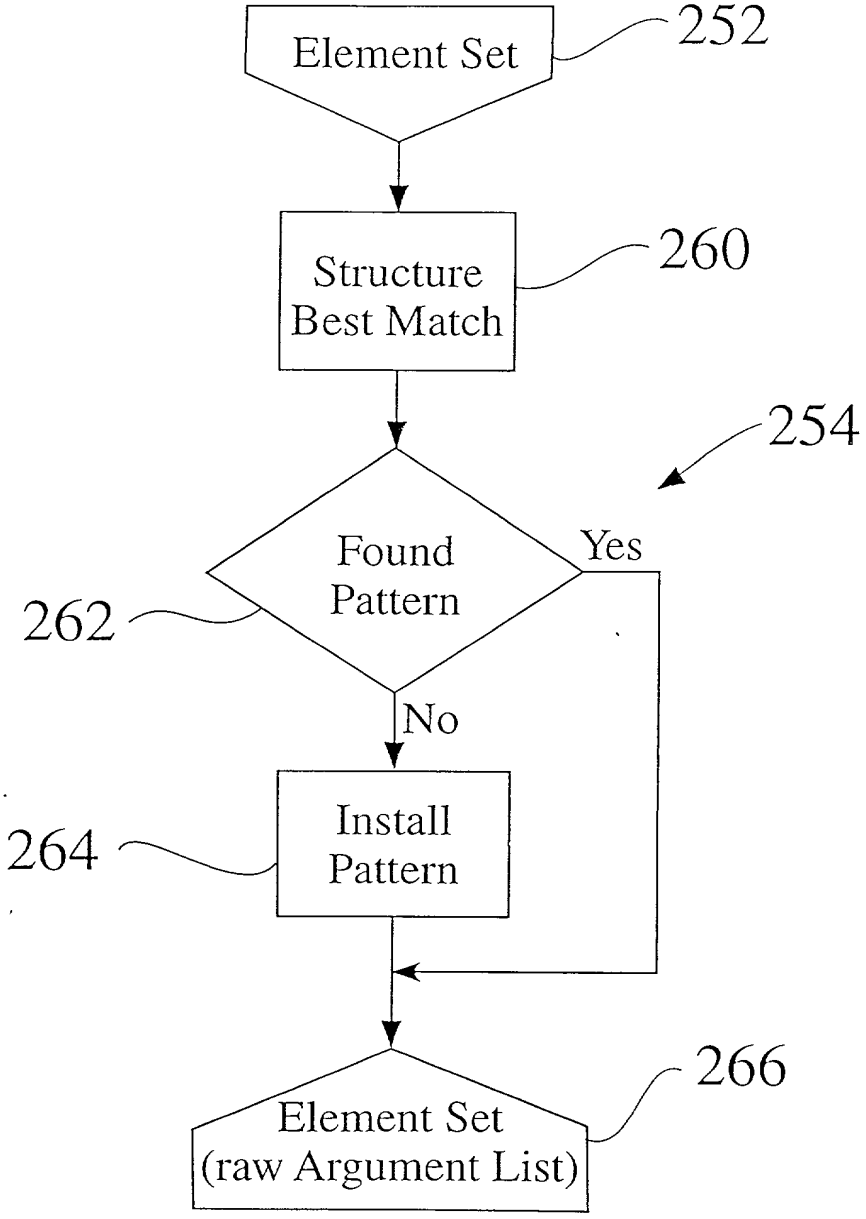

Element Set
252
Structure Best Match
260
254
Found Pattern
Yes
262
No
Install Pattern
264
Element Set (raw Argument List)
266

Fig. 13

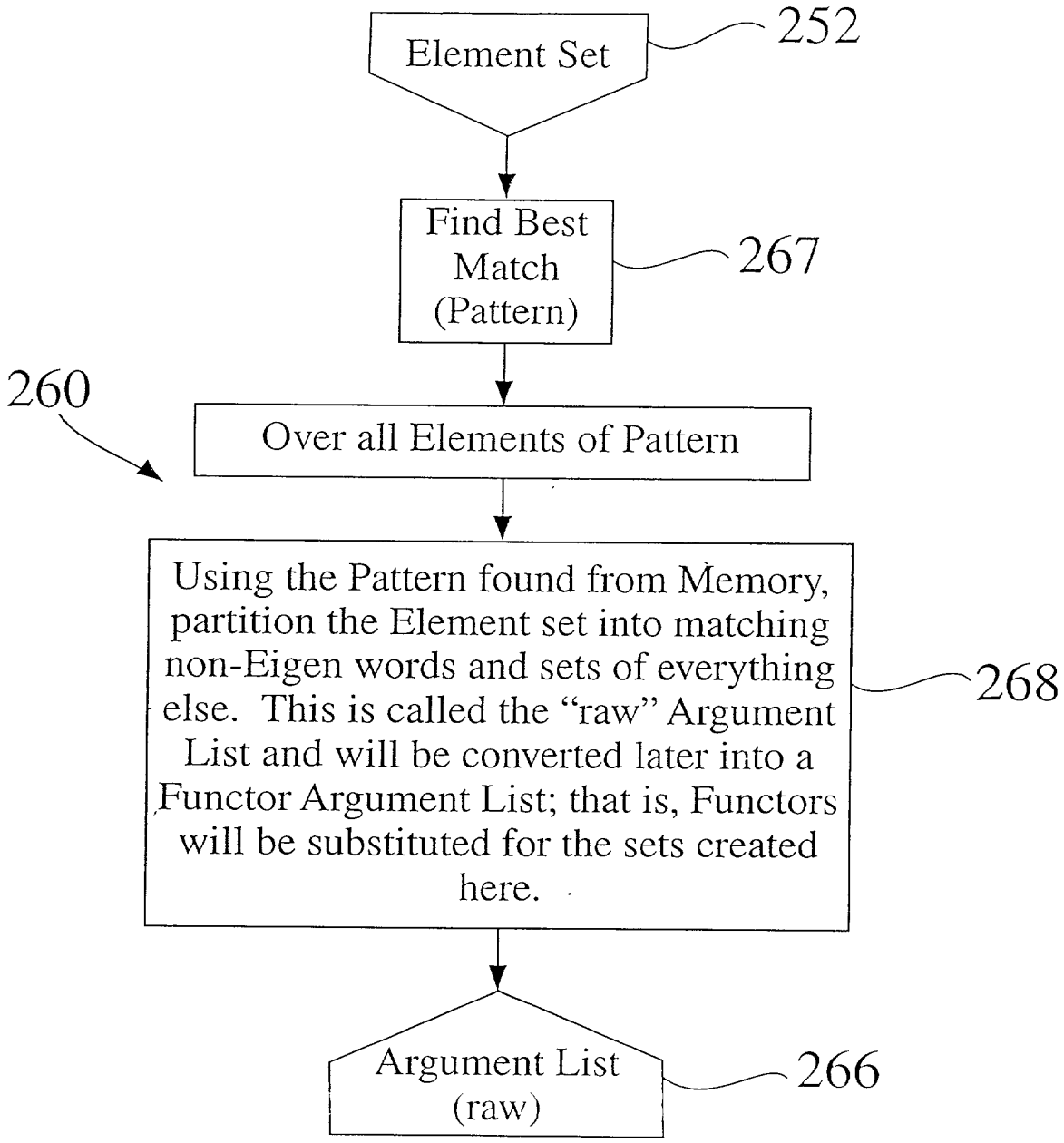

Element Set
252
Find Best Match (Pattern)
267
260
Over all Elements of Pattern
Using the Pattern found from Memory, partition the Element set into matching non-Eigen words and sets of everything else. This is called the "raw" Argument List and will be converted later into a Functor Argument List; that is, Functors will be substituted for the sets created here.
268
Argument List (raw)
266

Fig. 14

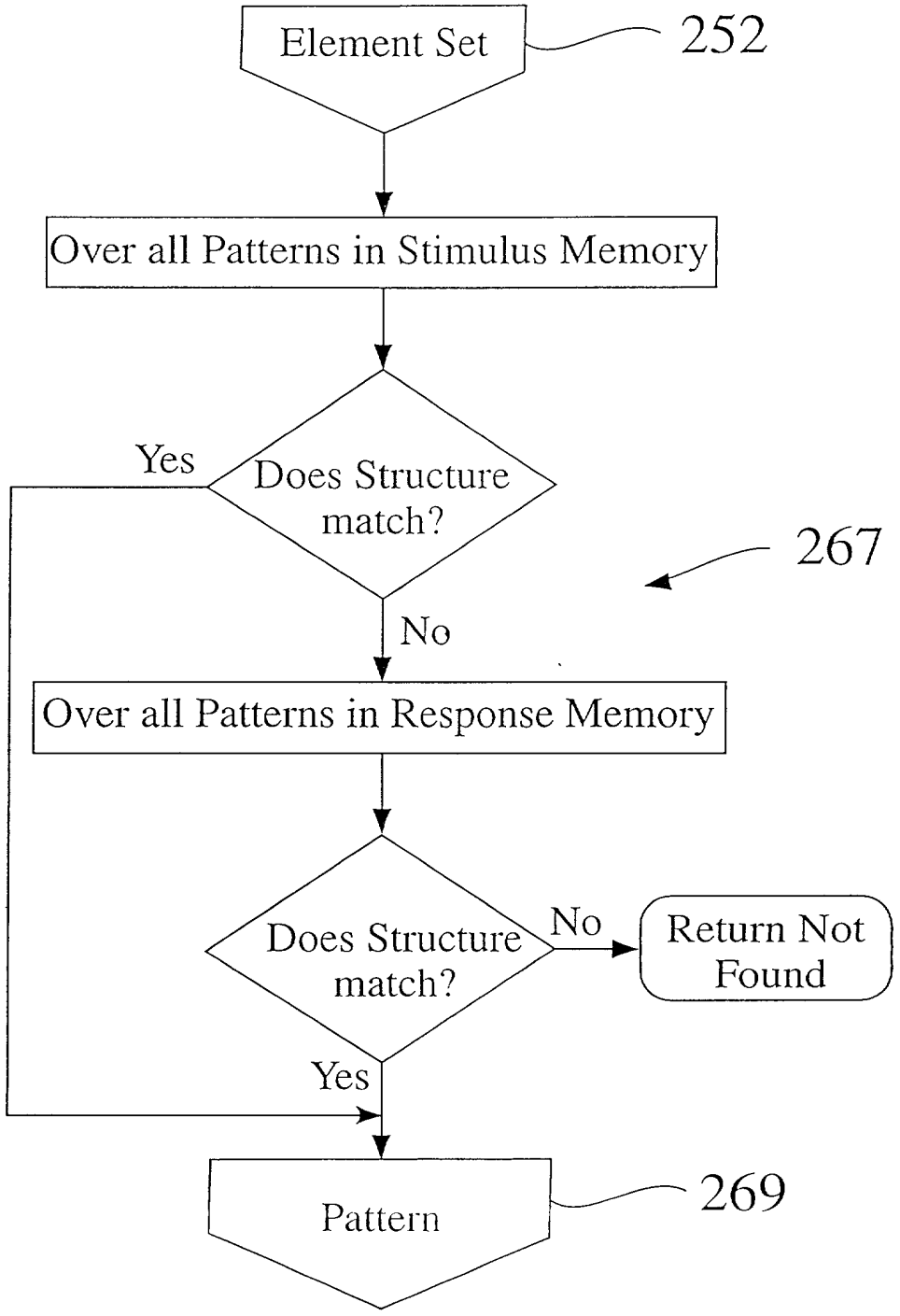

Element Set
252
Over all Patterns in Stimulus Memory
Yes
Does Structure match?
267
No
Over all Patterns in Response Memory
Does Structure match?
No
Return Not Found
Yes
Pattern
269

Fig. 15

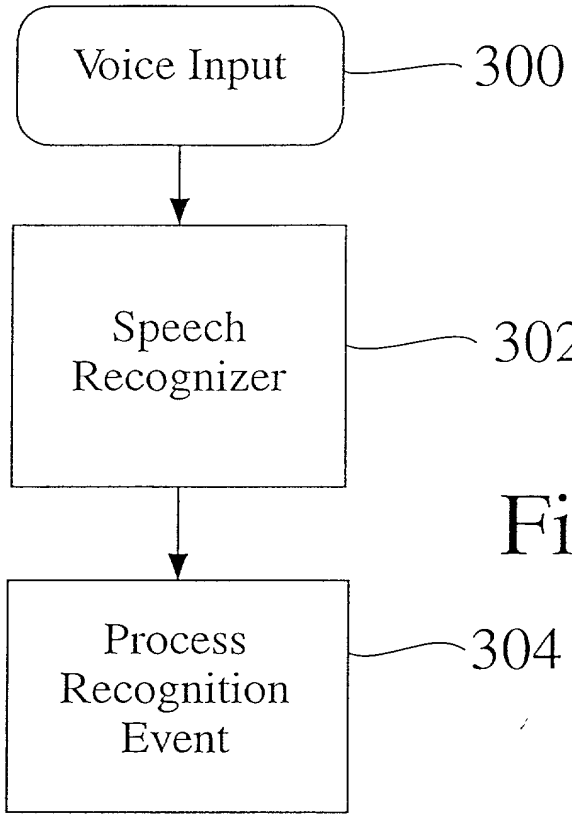

Voice Input
300
Speech Recognizer
302
Process Recognition Event
304

Fig. 16

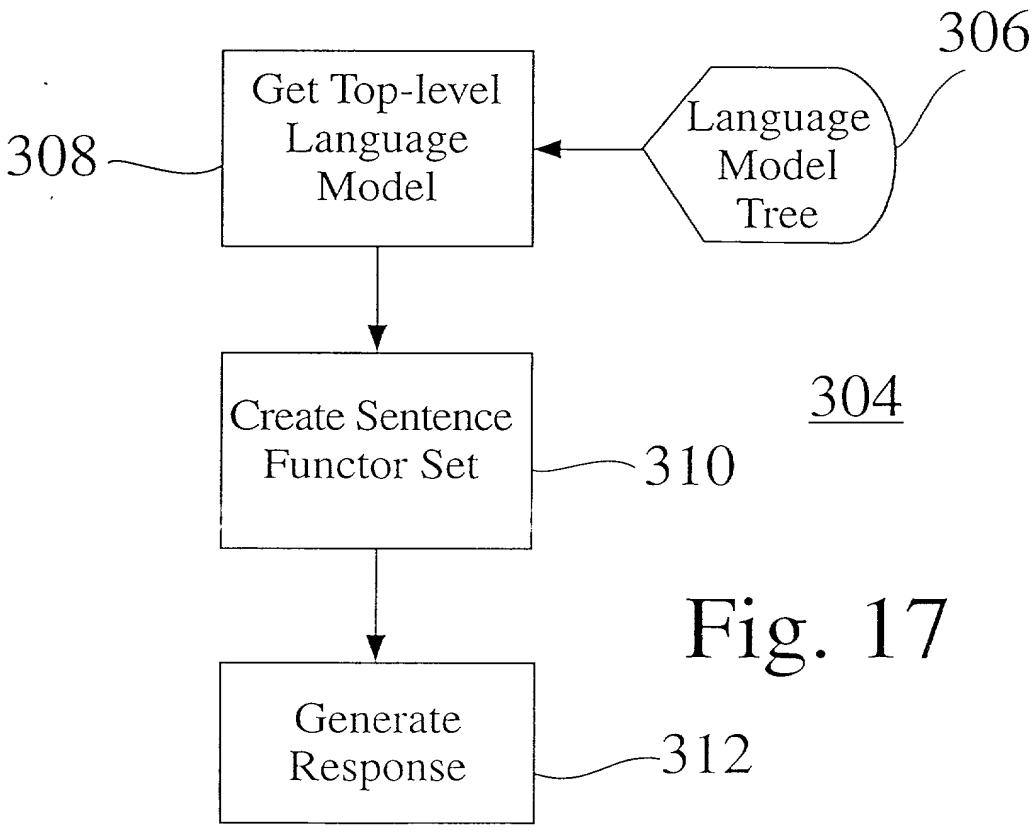

306
Get Top-level Language Model
308
Language Model Tree
Create Sentence Functor Set
310
304
Generate Response
312

Fig. 17

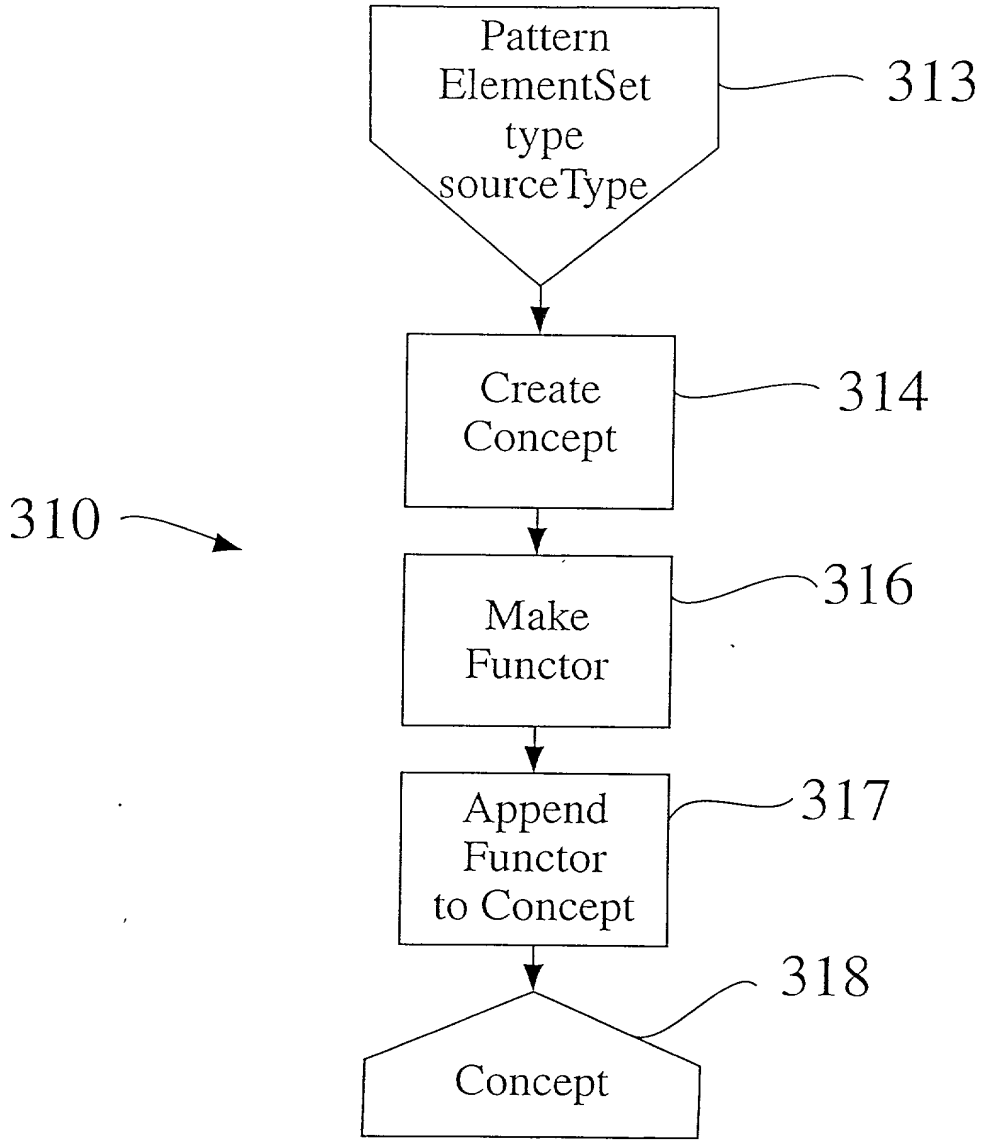

Pattern
ElementSet
type
sourceType
313
Create
Concept
314
310
Make
Functor
316
Append
Functor
to Concept
317
318
Concept

Fig. 18

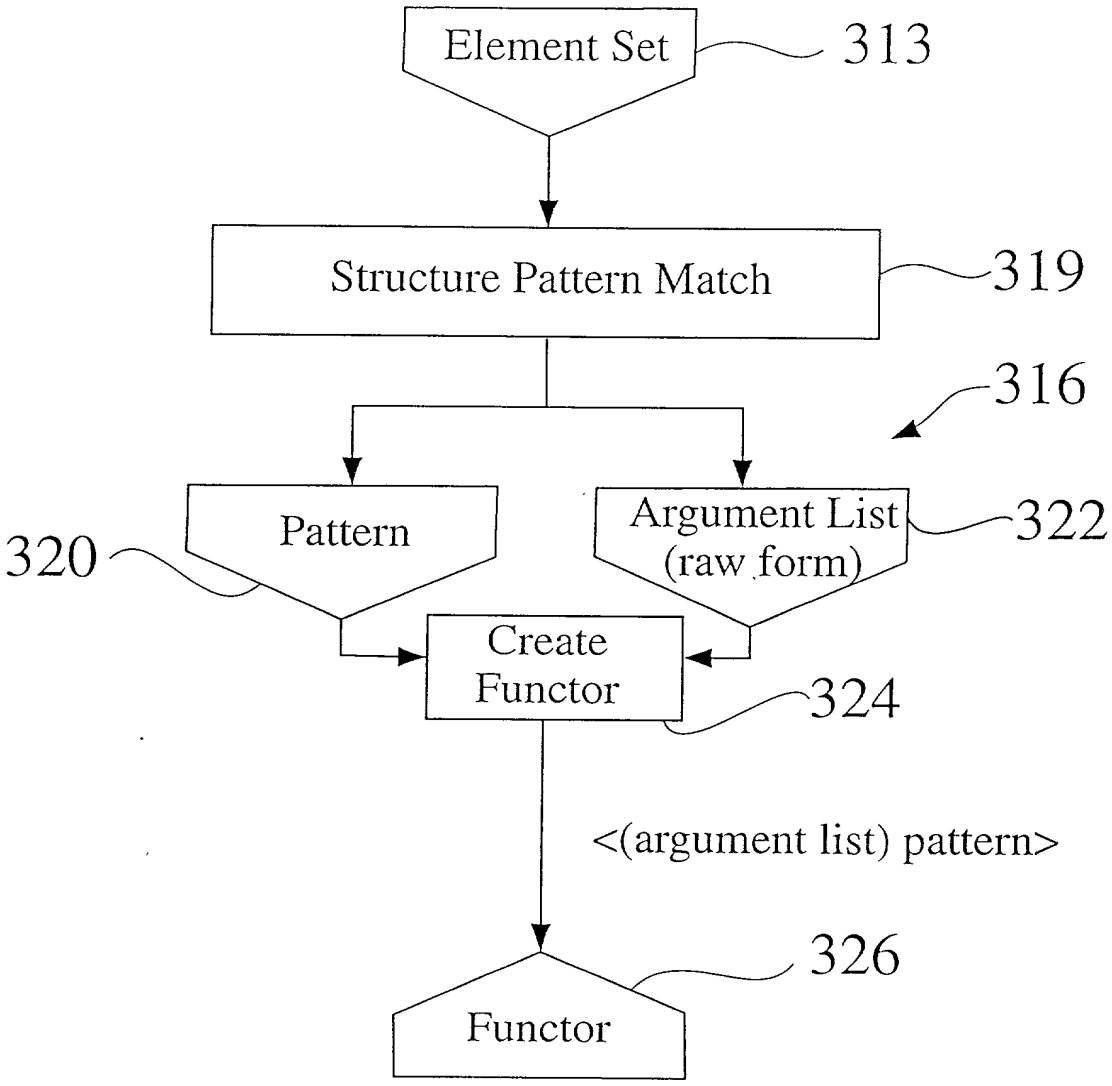

Element Set
313
Structure Pattern Match
319
316
Pattern
320
Argument List
(raw form)
322
Create
Functor
324
<(argument list) pattern>
326
Functor

Fig. 19

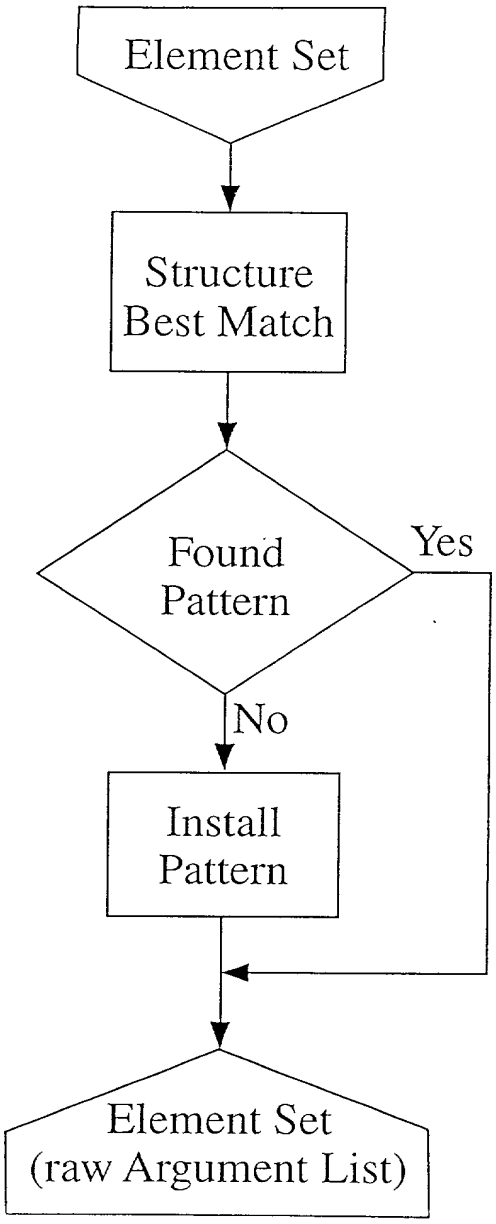

Element Set
Structure Best Match
Found Pattern
Yes
No
Install Pattern
Element Set (raw Argument List)

Fig. 19A

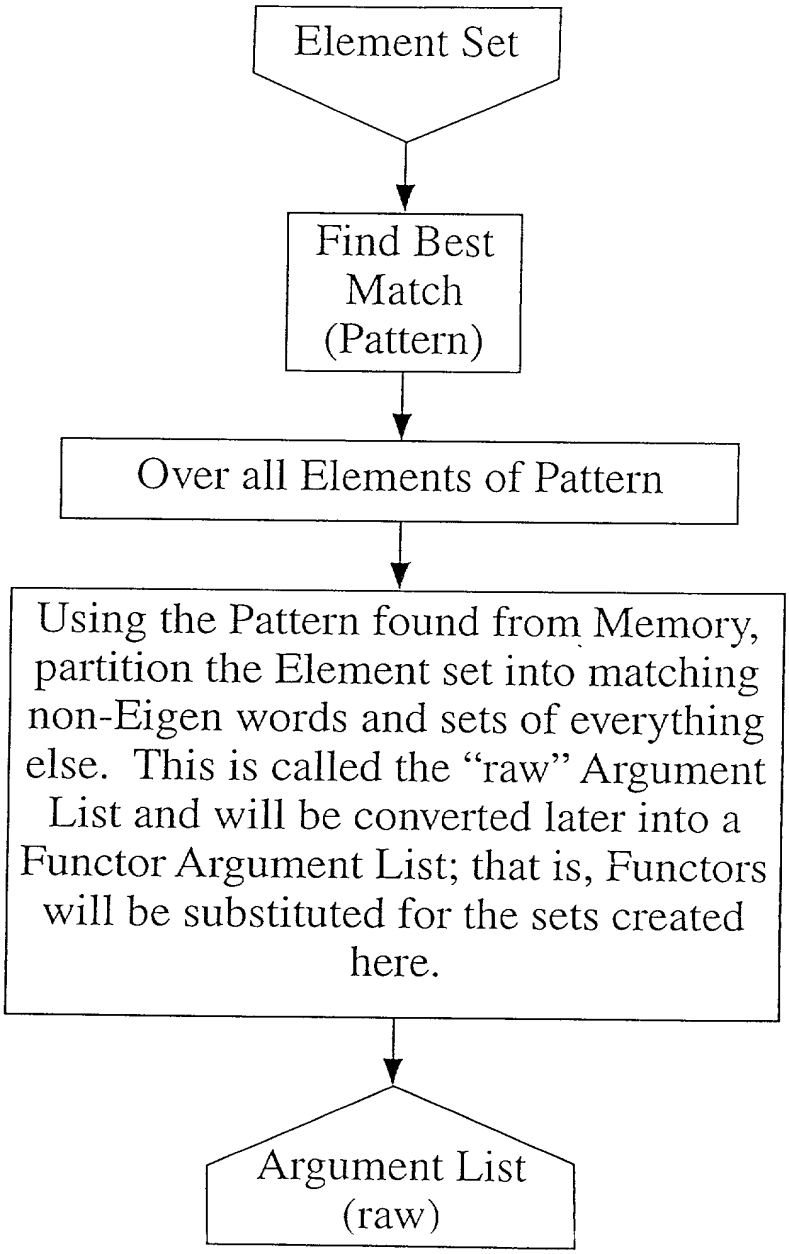

Element Set
Find Best Match (Pattern)
Over all Elements of Pattern
Using the Pattern found from Memory, partition the Element set into matching non-Eigen words and sets of everything else. This is called the "raw" Argument List and will be converted later into a Functor Argument List; that is, Functors will be substituted for the sets created here.
Argument List (raw)

Fig. 19B

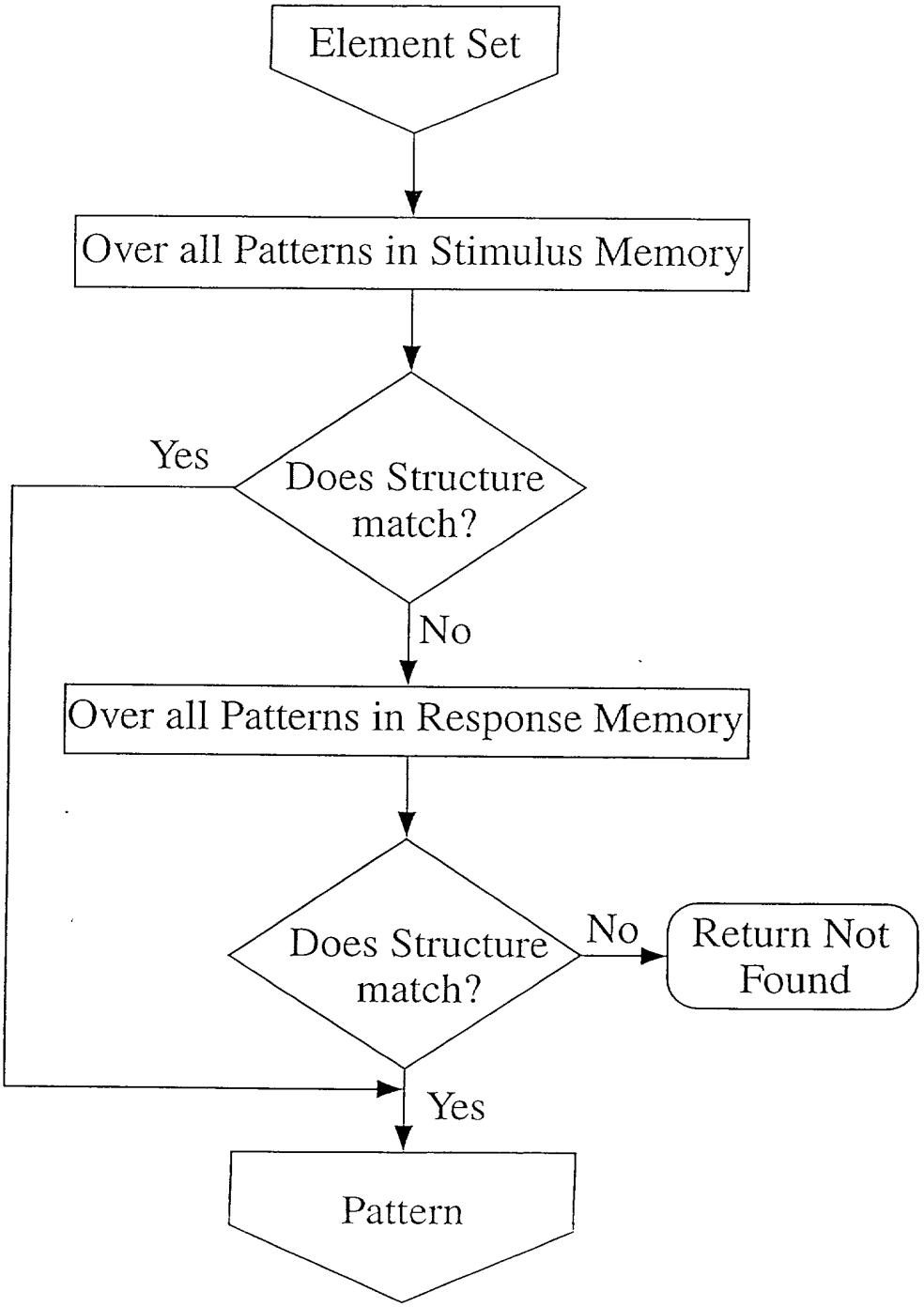

Element Set
Over all Patterns in Stimulus Memory
Does Structure match?
Yes
No
Over all Patterns in Response Memory
Does Structure match?
No
Return Not Found
Yes
Pattern

Fig. 19C

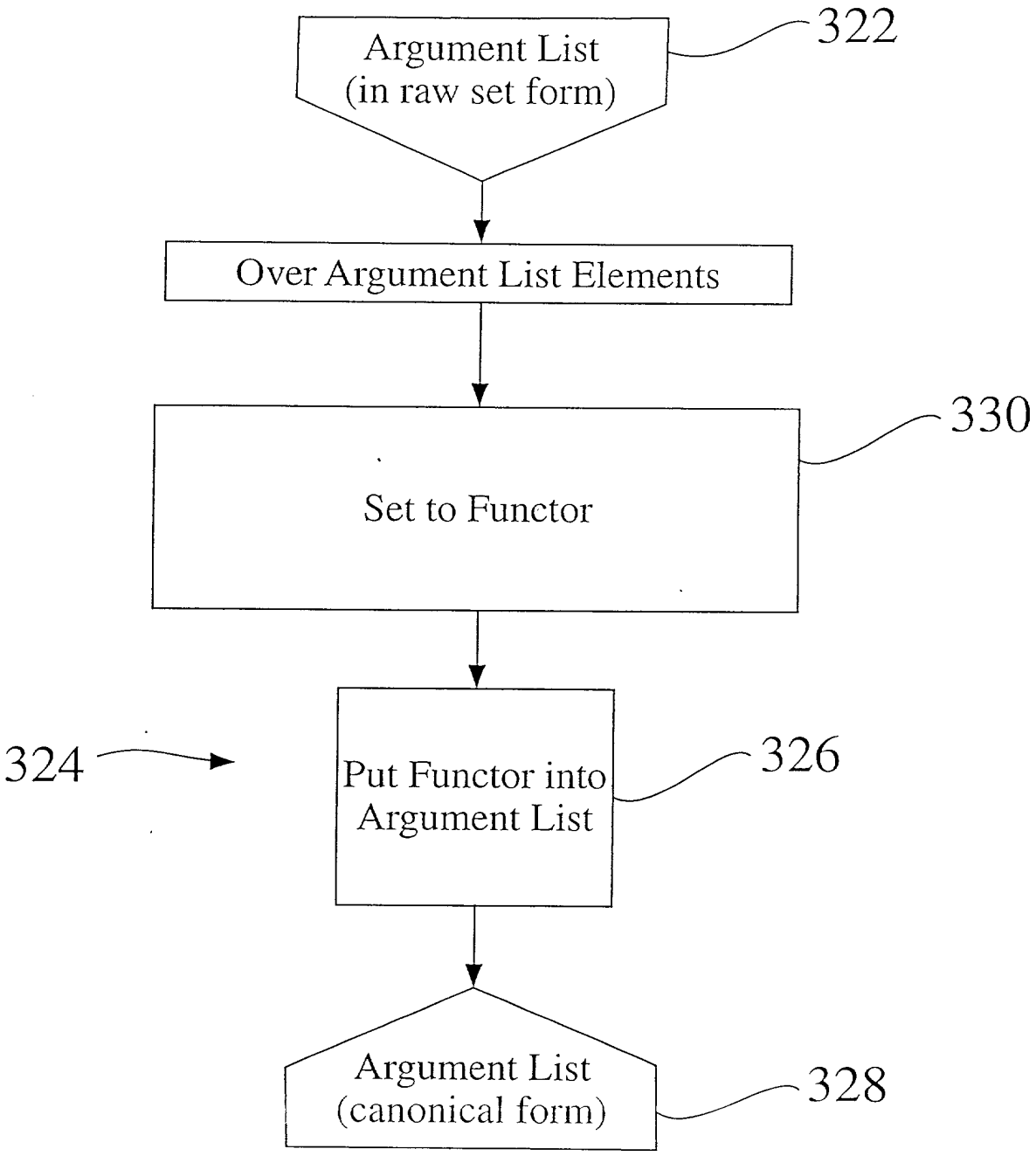

Argument List
(in raw set form)
322
Over Argument List Elements
330
Set to Functor
324
Put Functor into
Argument List
326
Argument List
(canonical form)
328

Fig. 20

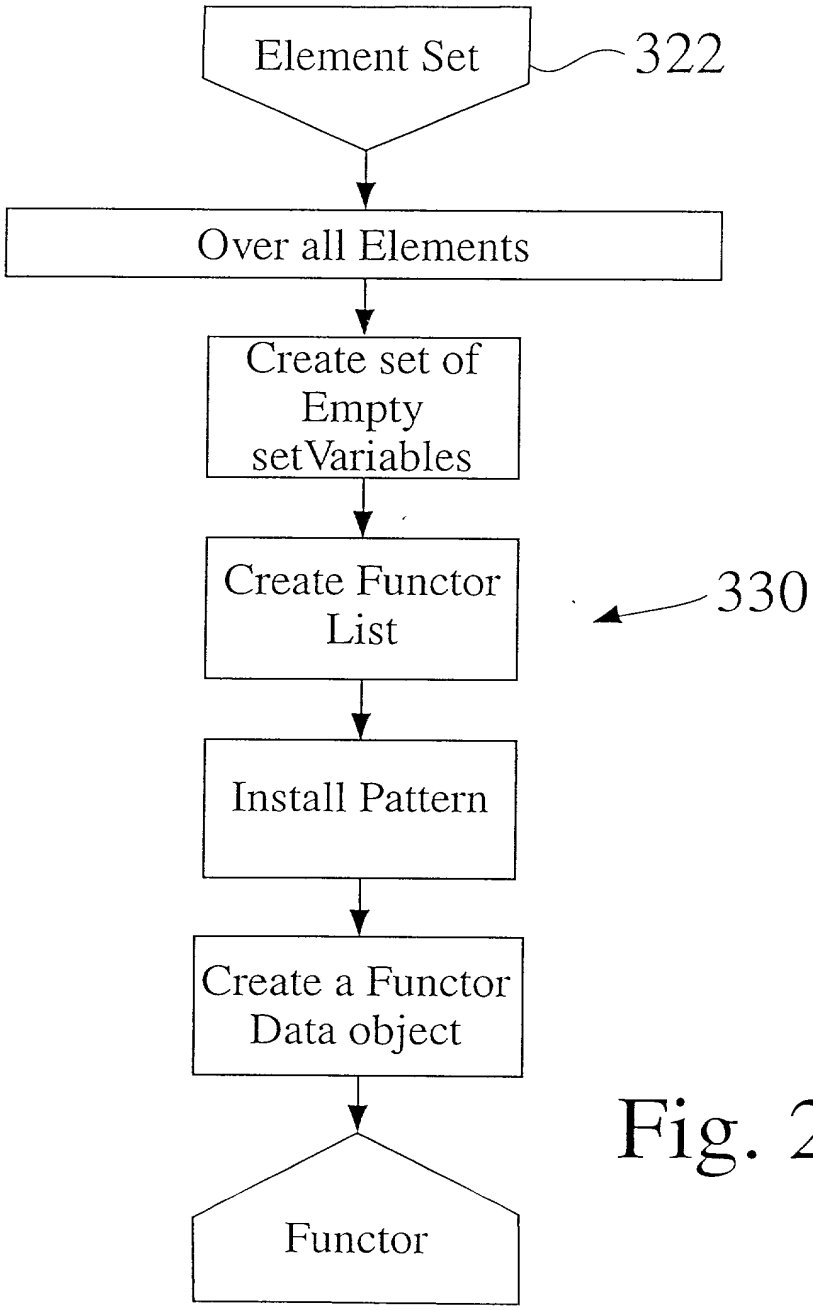

Element Set
322
Over all Elements
Create set of Empty setVariables
Create Functor List
330
Install Pattern
Create a Functor Data object
Functor

Fig. 21

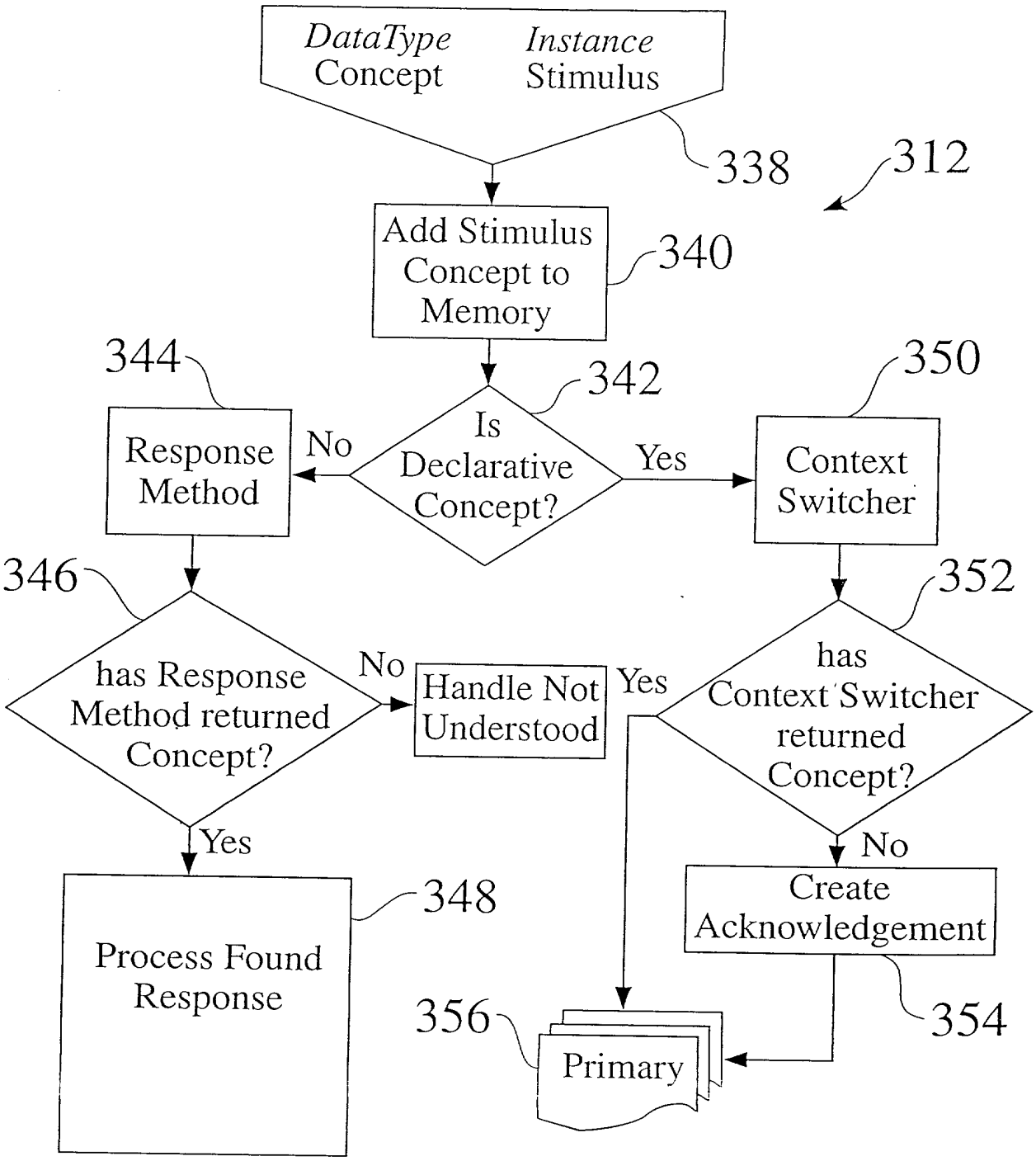

DataType Concept
Instance Stimulus
338
312
Add Stimulus Concept to Memory
340
344
342
350
Response Method
No
Is Declarative Concept?
Yes
Context Switcher
346
352
has Response Method returned Concept?
No
Handle Not Understood
Yes
has Context Switcher returned Concept?
Yes
No
Process Found Response
348
Create Acknowledgement
356
354
Primary

Fig. 22

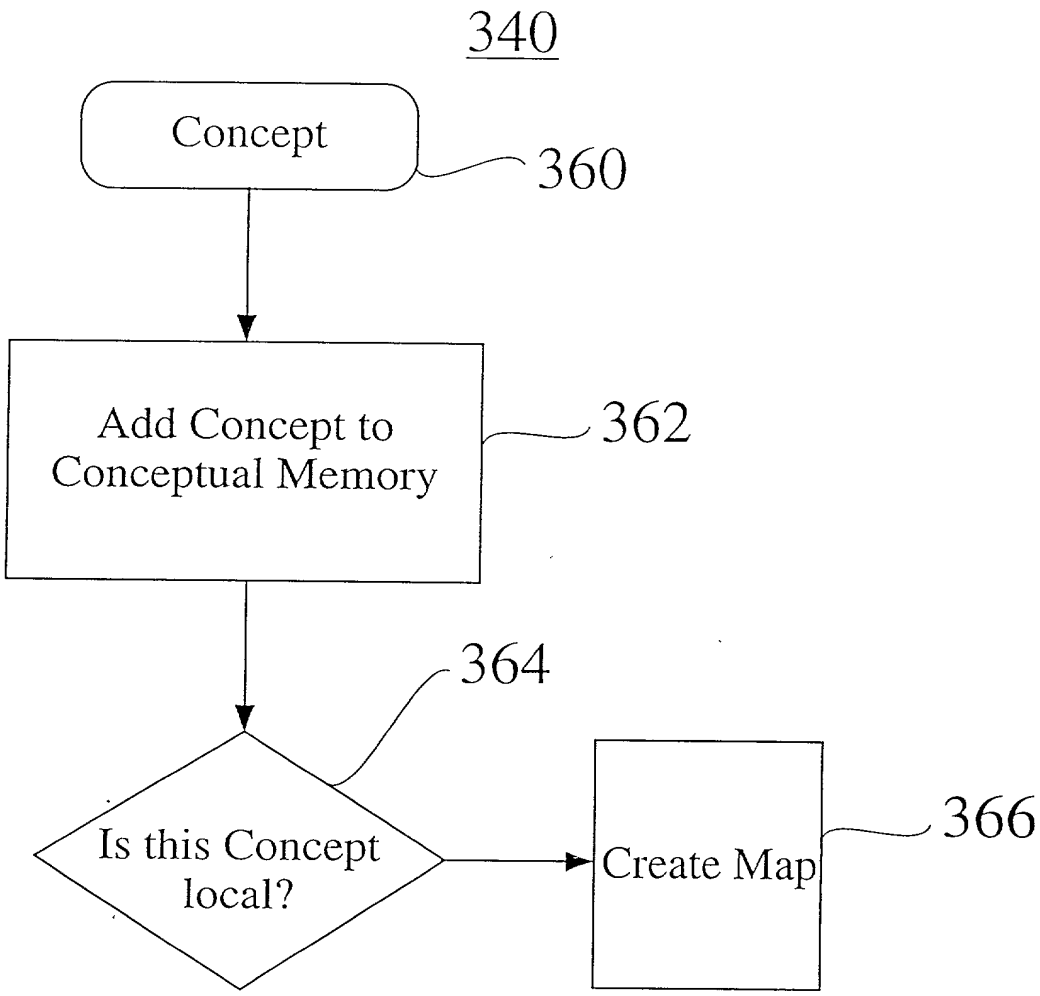

340
Concept
360
Add Concept to Conceptual Memory
362
364
Is this Concept local?
Create Map
366

Fig. 23

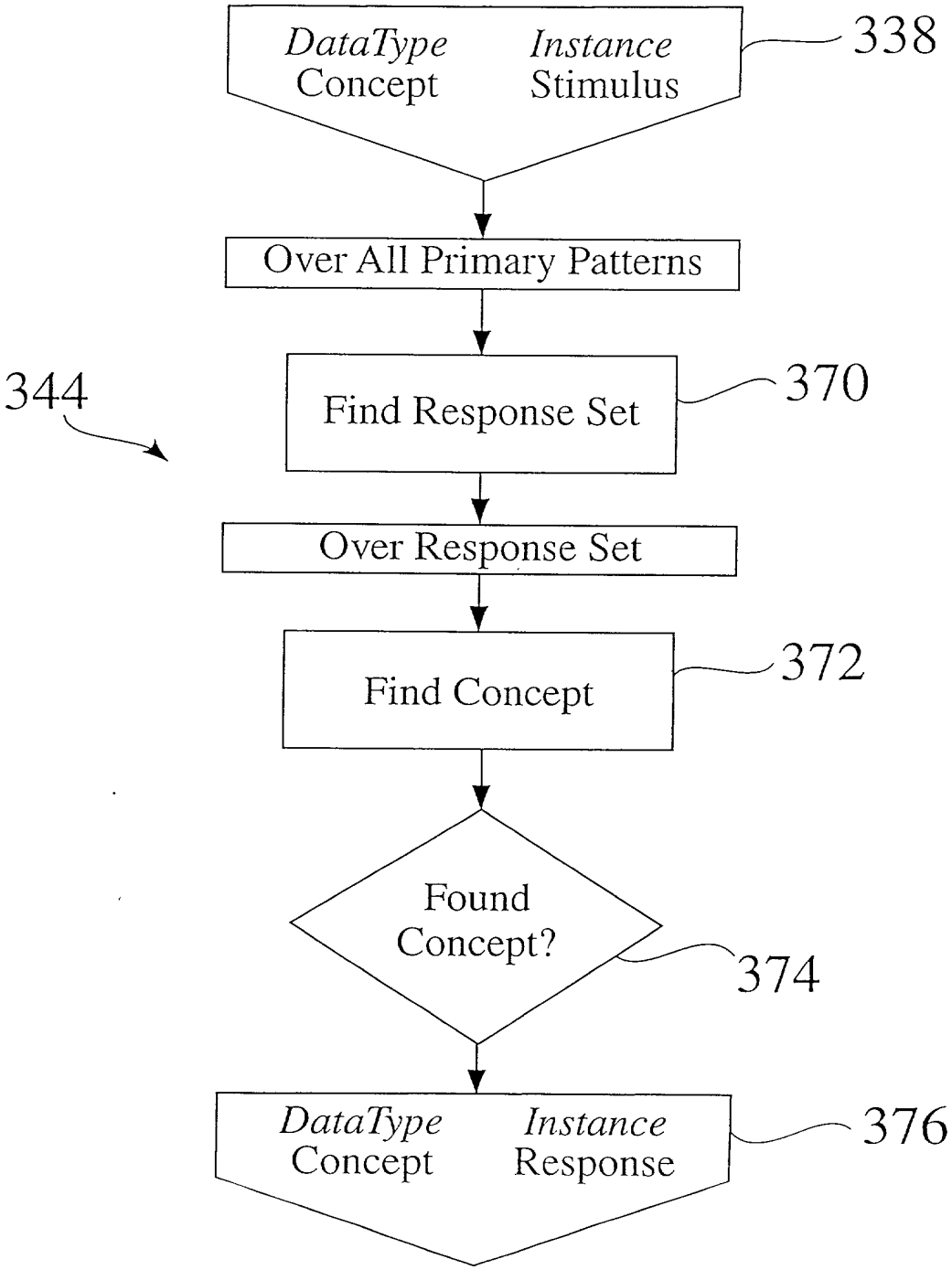

DataType Concept
Instance Stimulus
338
Over All Primary Patterns
344
Find Response Set
370
Over Response Set
Find Concept
372
Found Concept?
374
DataType Concept
Instance Response
376

Fig. 24

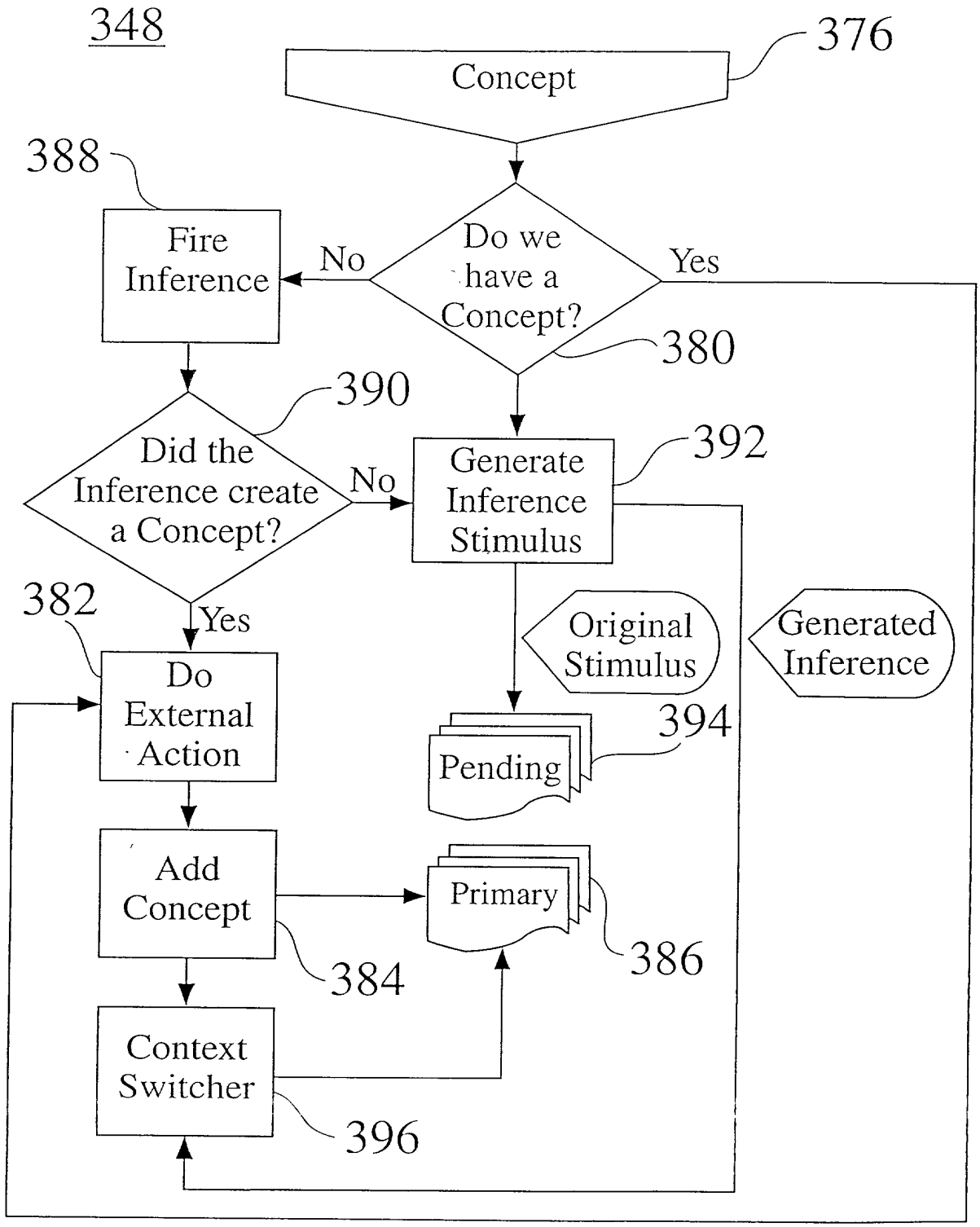

348
376
Concept
388
Fire Inference
No
Do we have a Concept?
Yes
380
390
Did the Inference create a Concept?
No
Generate Inference Stimulus
392
382
Yes
Original Stimulus
Generated Inference
Do External Action
394
Pending
Add Concept
Primary
384
386
Context Switcher
396

Fig. 25

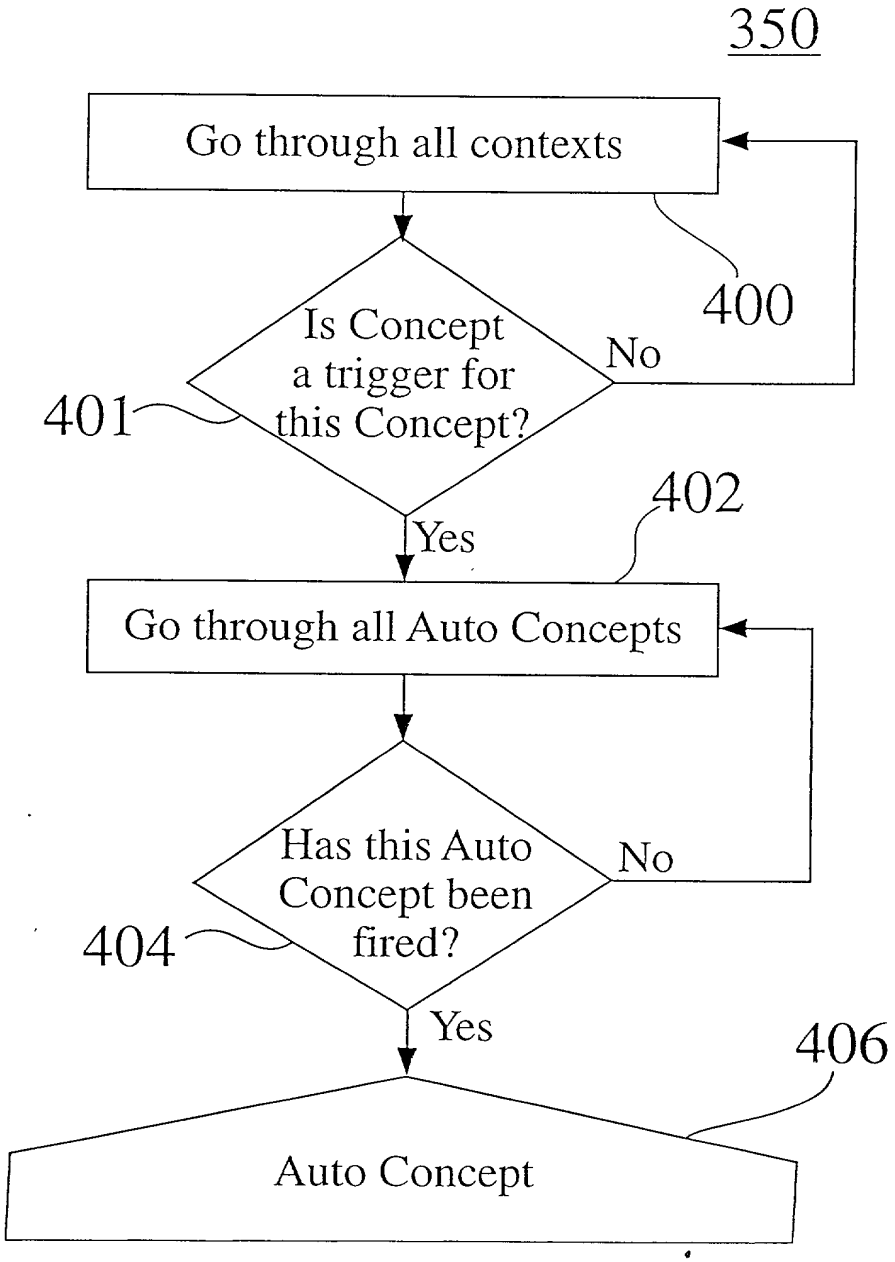

350
Go through all contexts
400
Is Concept a trigger for this Concept?
401
No
Yes
402
Go through all Auto Concepts
Has this Auto Concept been fired?
404
No
Yes
406
Auto Concept

Fig. 26

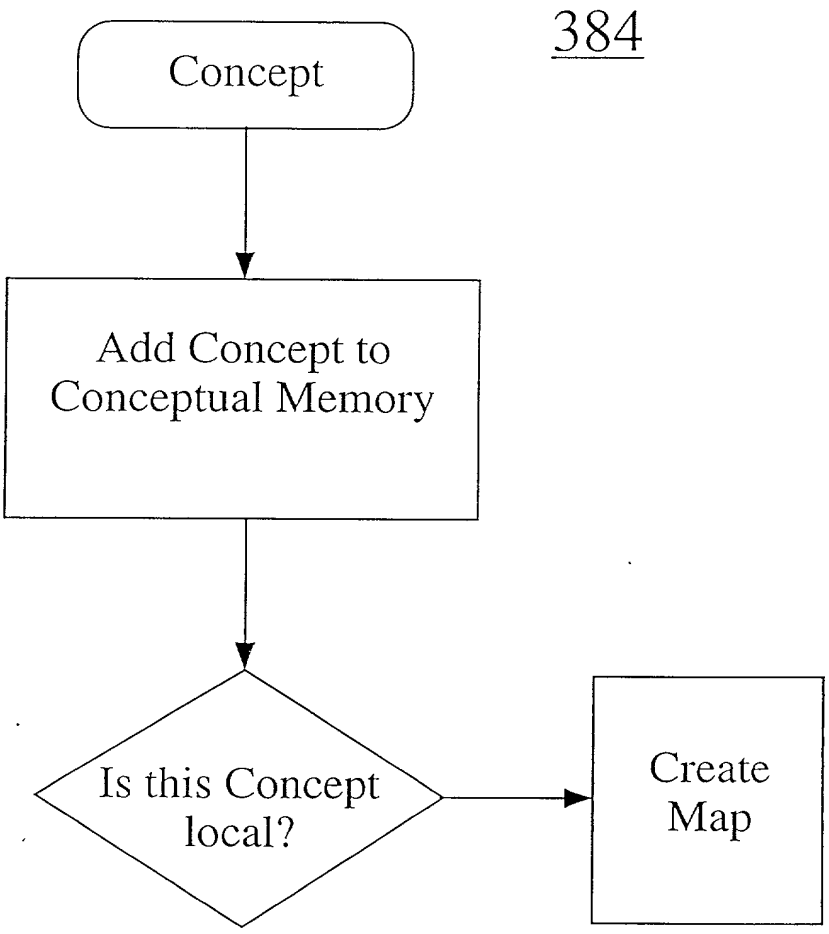

384
Concept
Add Concept to Conceptual Memory
Is this Concept local?
Create Map

Fig. 26A

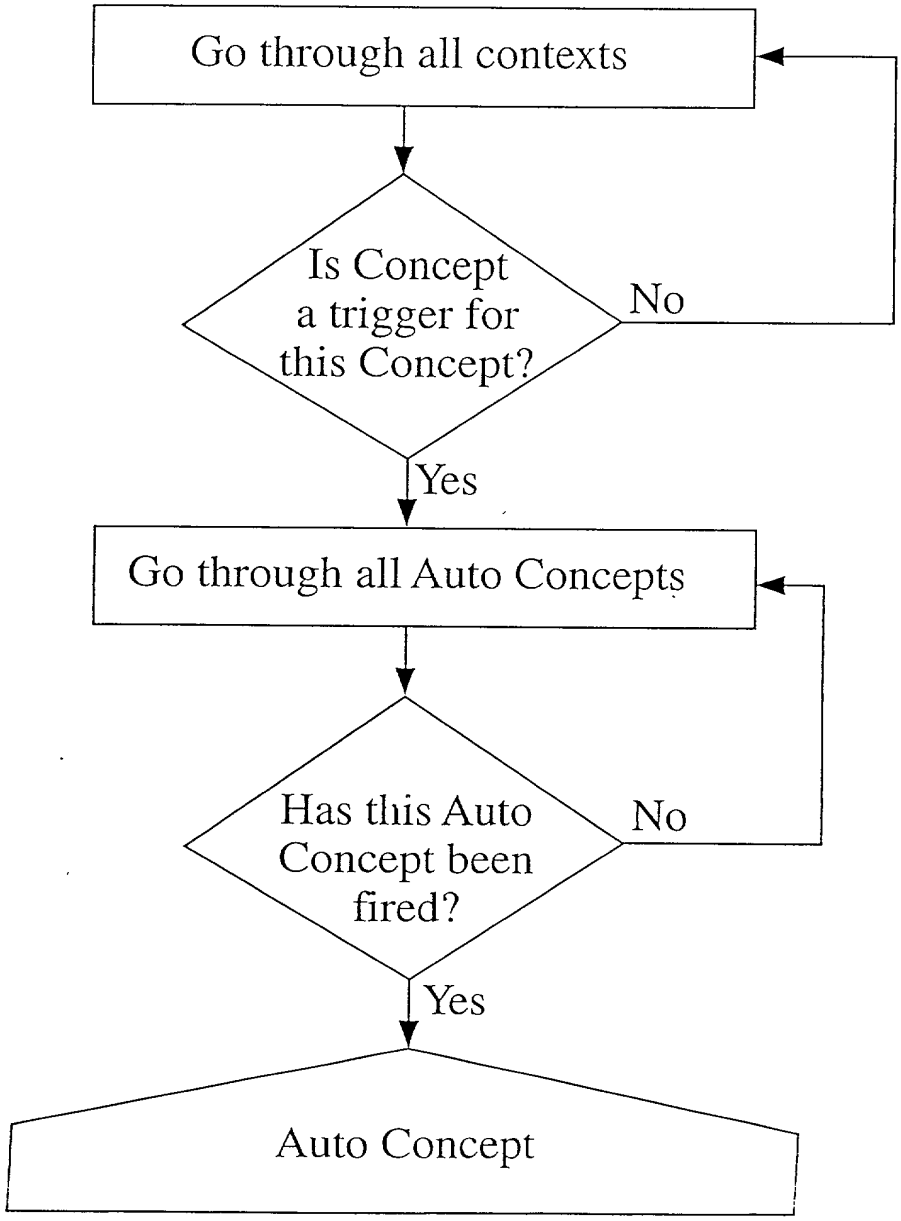

Go through all contexts
Is Concept a trigger for this Concept?
No
Yes
Go through all Auto Concepts
Has this Auto Concept been fired?
No
Yes
Auto Concept

Fig. 26B

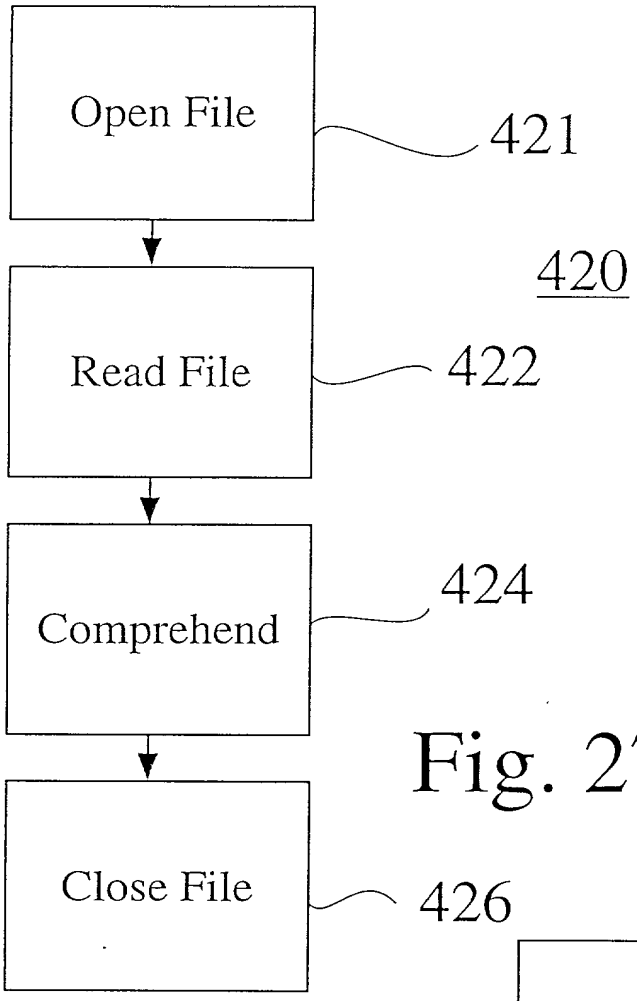

Open File
421
420
Read File
422
Comprehend
424
Close File
426

Fig. 27

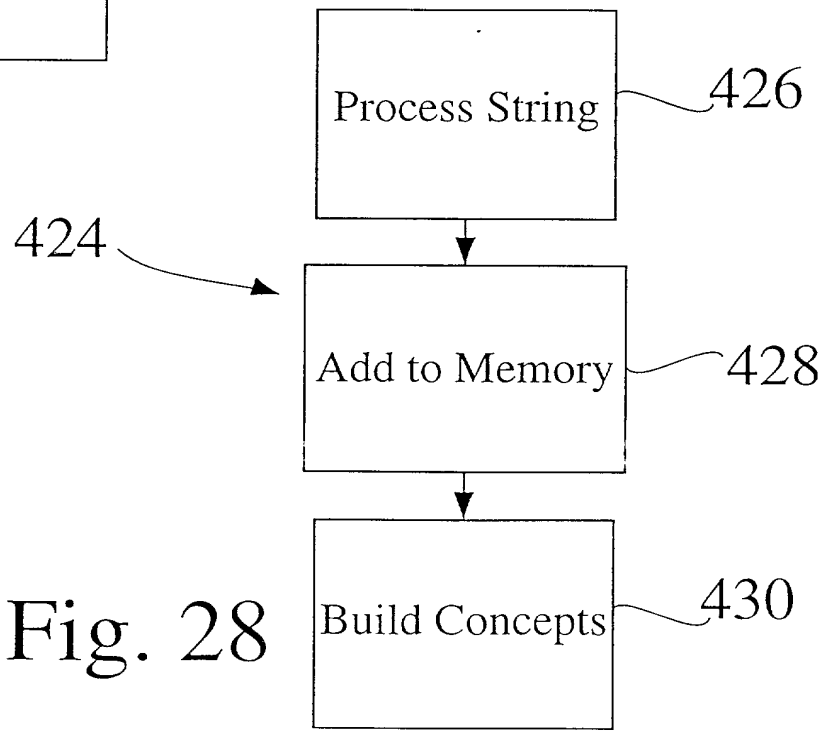

Process String
426
424
Add to Memory
428
Build Concepts
430

Fig. 28

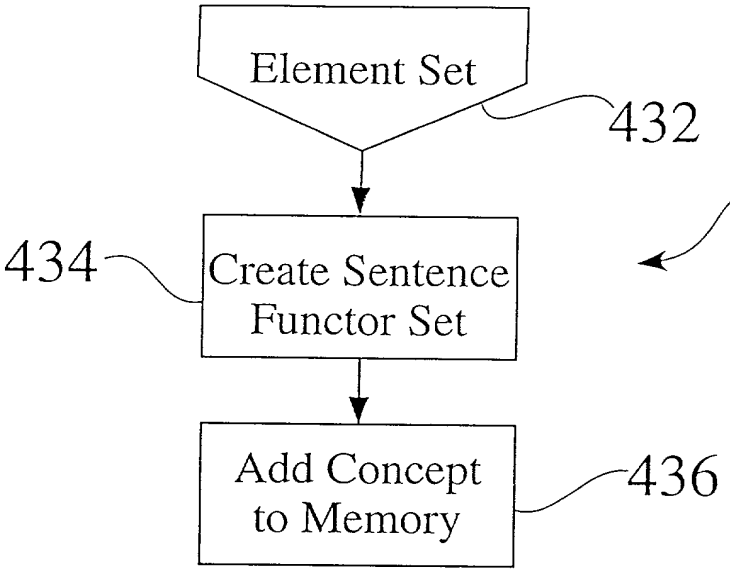

Element Set
432
430
434
Create Sentence Functor Set
Add Concept to Memory
436

Fig. 29

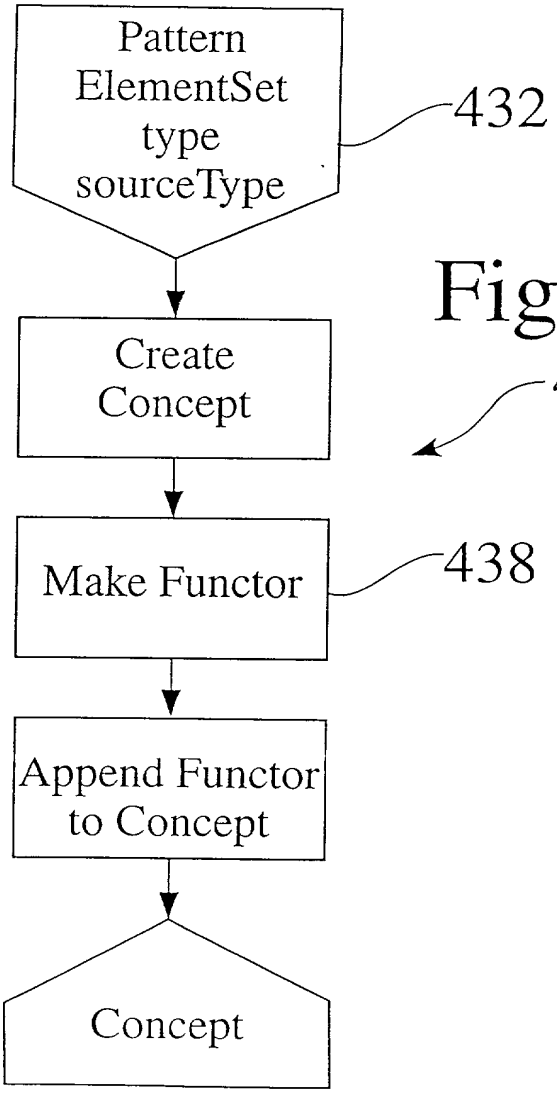

Pattern
ElementSet
type
sourceType
432
Create Concept
430
Make Functor
438
Append Functor to Concept
Concept

Fig. 30

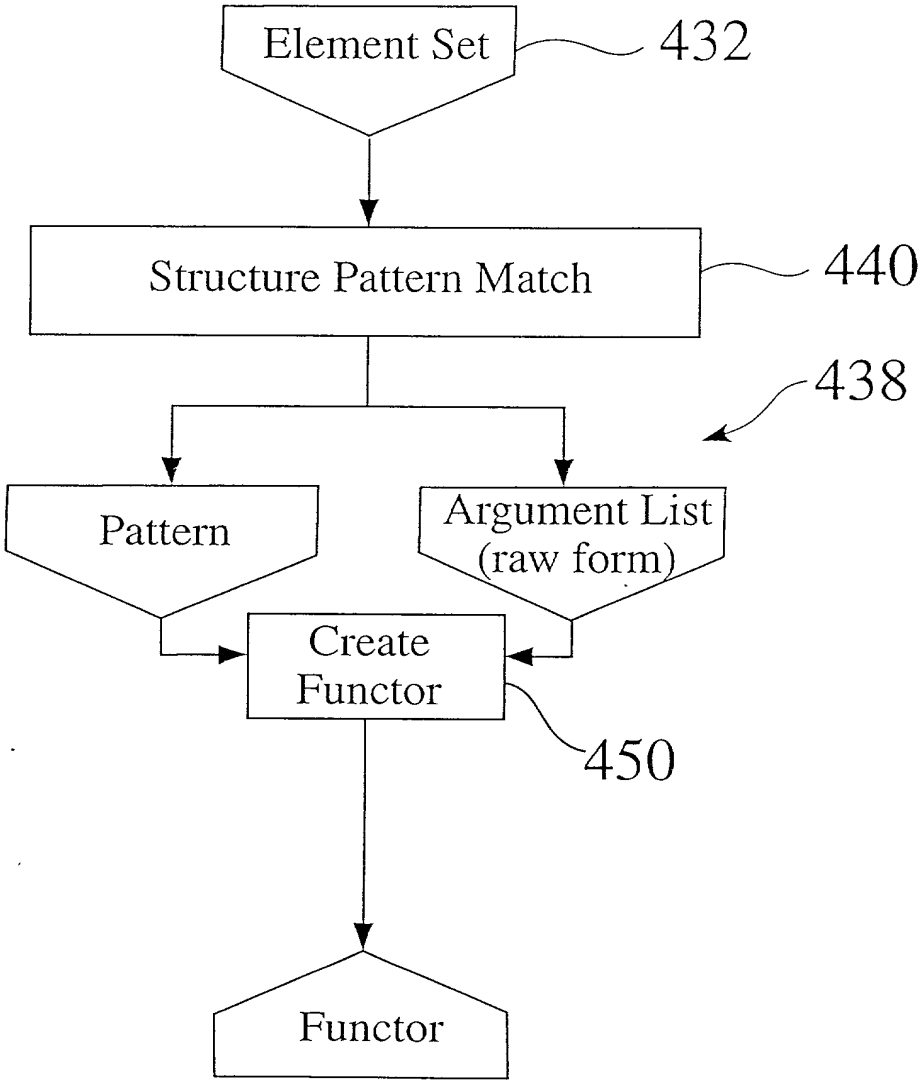

Element Set
432
Structure Pattern Match
440
438
Pattern
Argument List
(raw form)
Create
Functor
450
Functor

Fig. 31

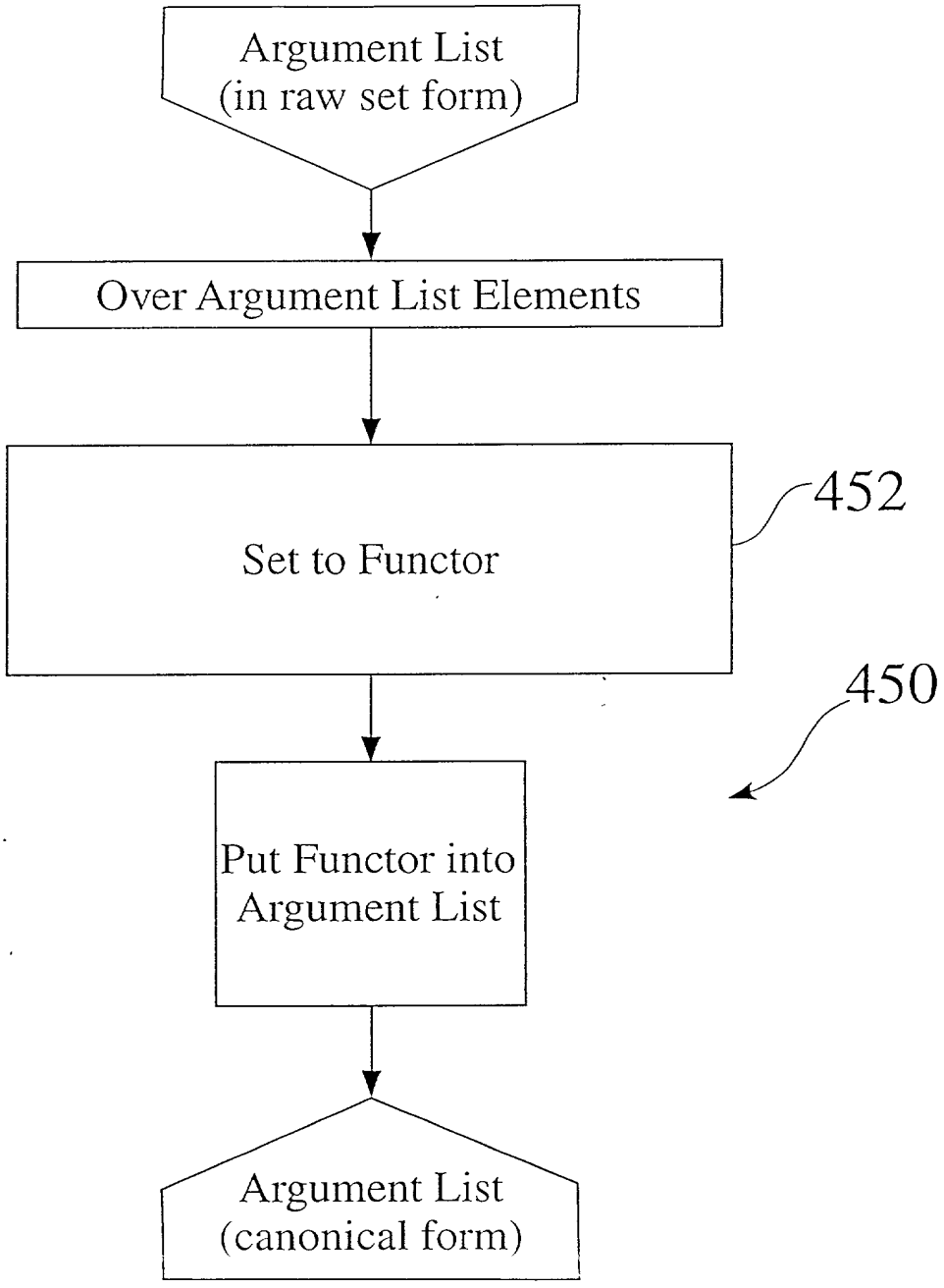

Argument List
(in raw set form)
Over Argument List Elements
Set to Functor
452
450
Put Functor into
Argument List
Argument List
(canonical form)

Fig. 32

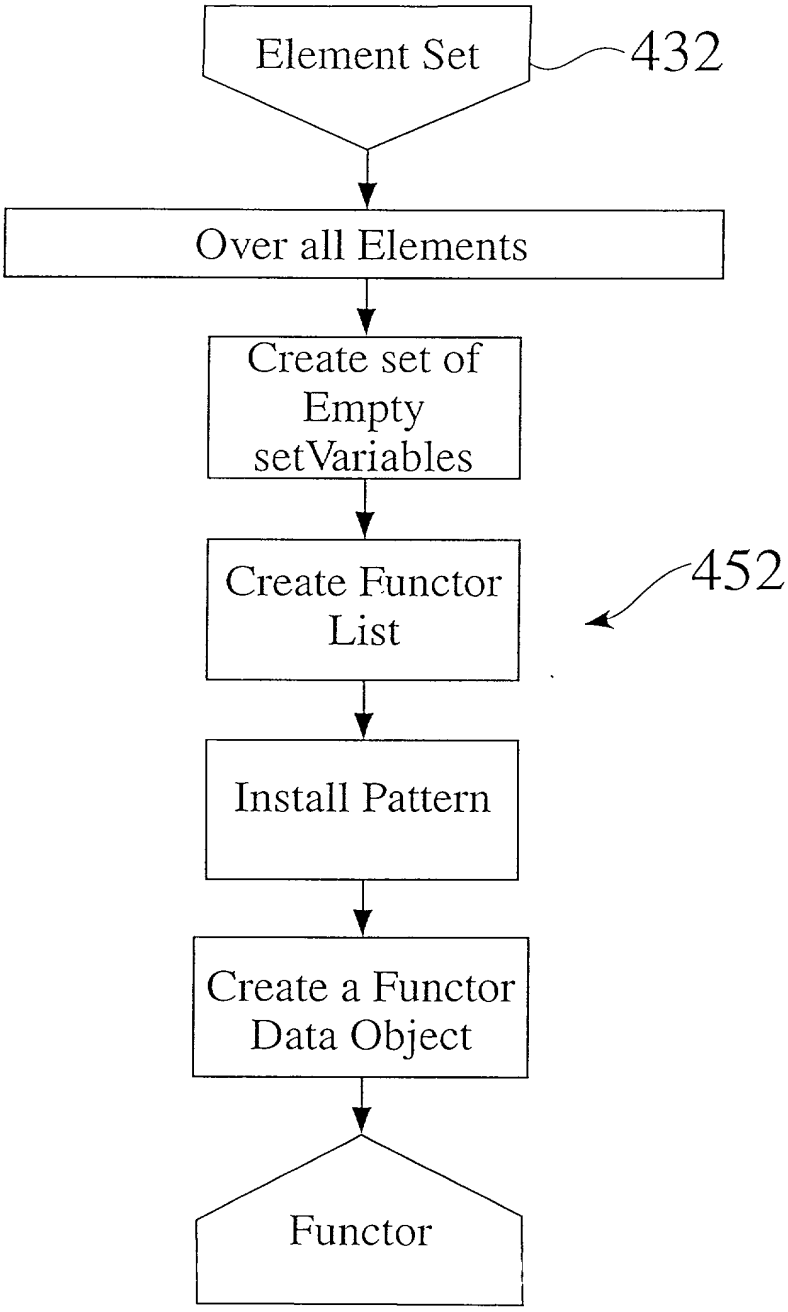

Element Set
432
Over all Elements
Create set of Empty setVariables
Create Functor List
452
Install Pattern
Create a Functor Data Object
Functor

Fig. 33

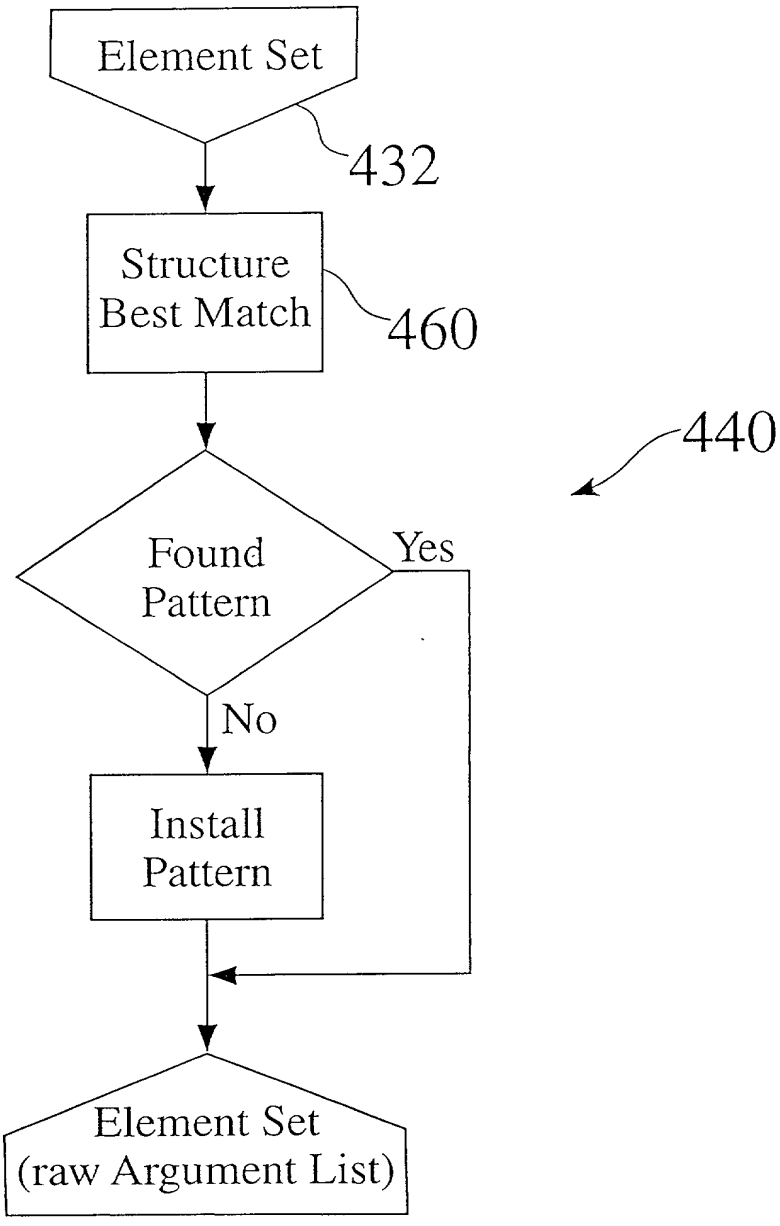

Element Set
432
Structure
Best Match
460
440
Found
Pattern
Yes
No
Install
Pattern
Element Set
(raw Argument List)

Fig. 34

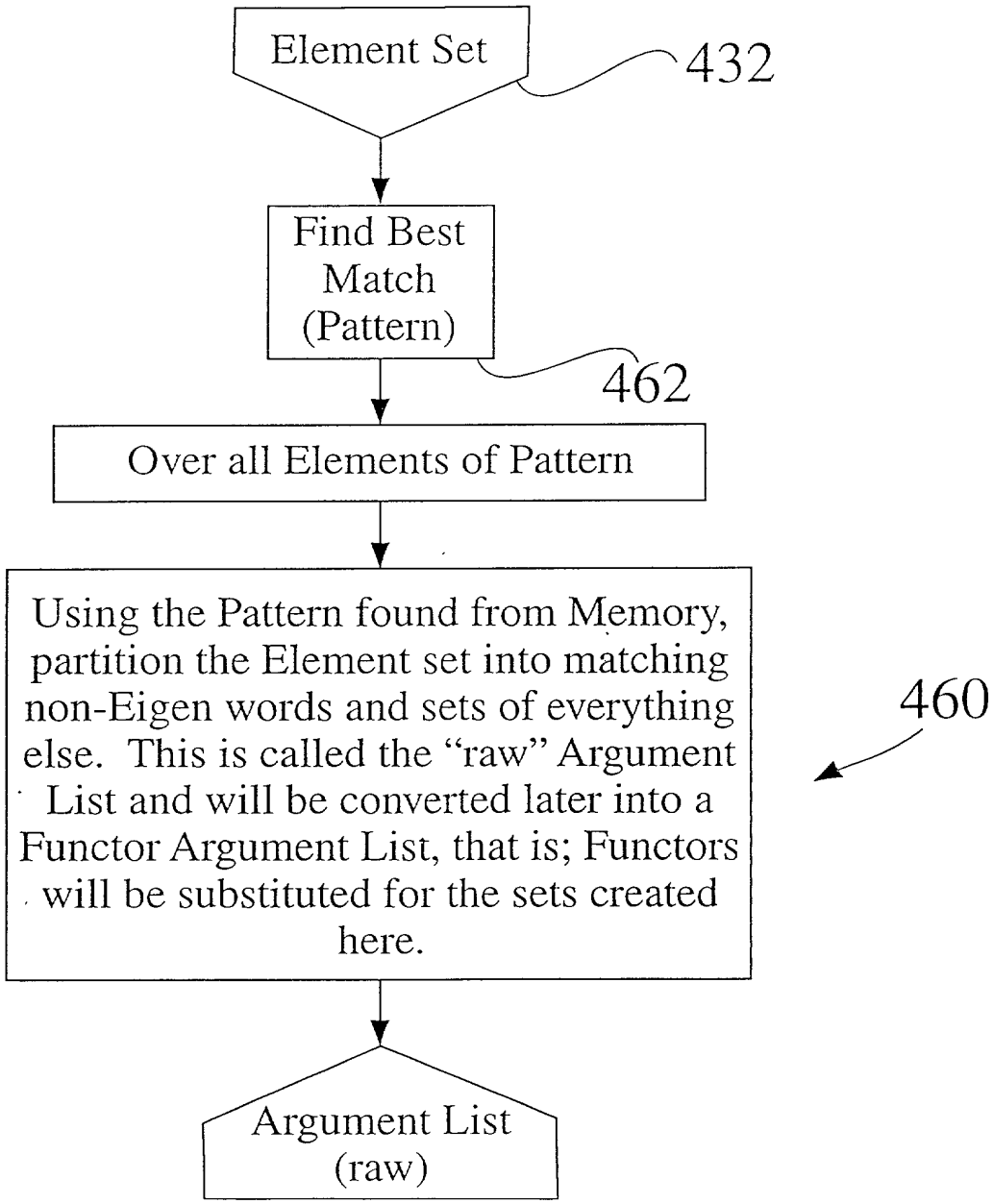

Element Set
432
Find Best Match (Pattern)
462
Over all Elements of Pattern
Using the Pattern found from Memory, partition the Element set into matching non-Eigen words and sets of everything else. This is called the "raw" Argument List and will be converted later into a Functor Argument List, that is; Functors will be substituted for the sets created here.
460
Argument List (raw)

Fig. 35

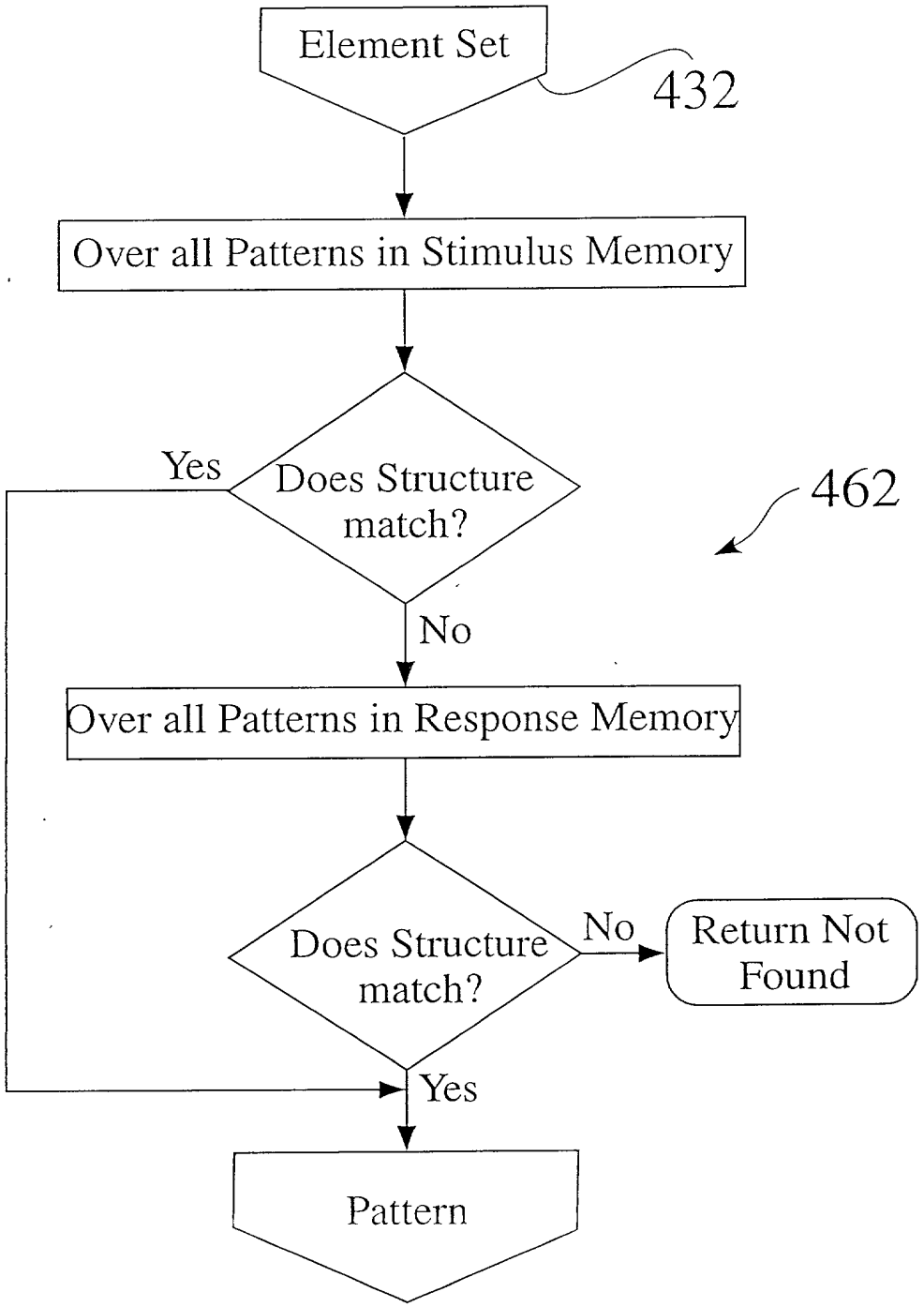

Element Set
432
Over all Patterns in Stimulus Memory
Yes
Does Structure match?
462
No
Over all Patterns in Response Memory
Does Structure match?
No
Return Not Found
Yes
Pattern

Fig. 36

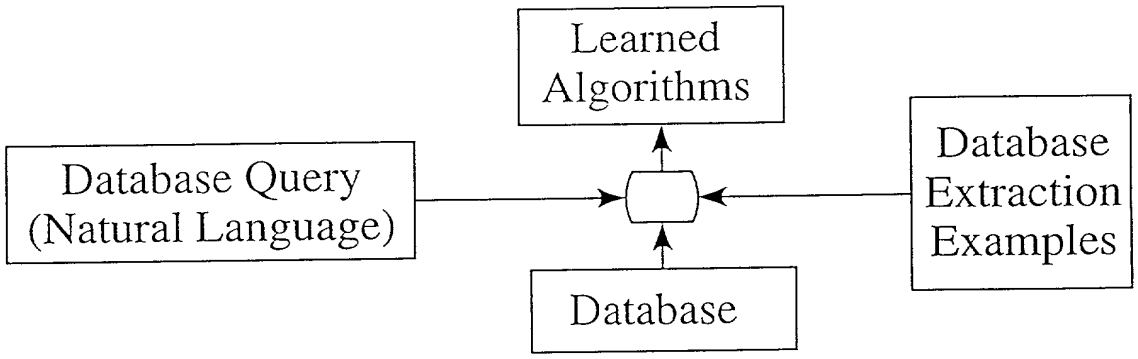

Learned Algorithms
Database Query (Natural Language)
Database Extraction Examples
Database

Fig. 37

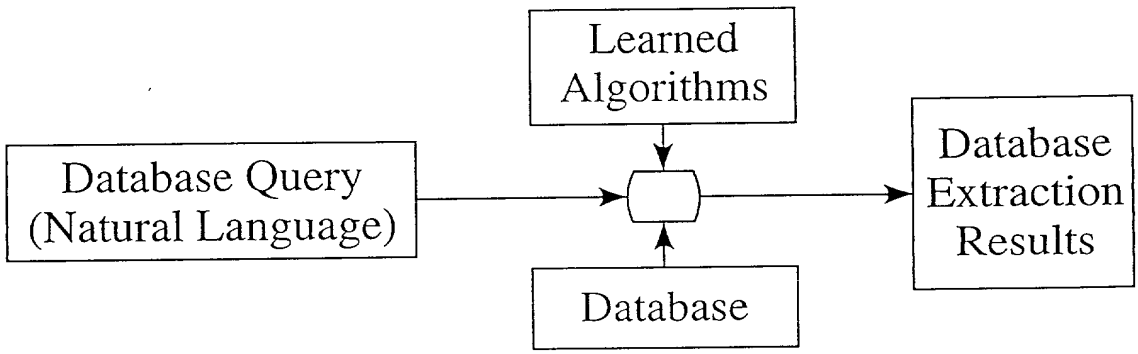

Learned Algorithms
Database Query (Natural Language)
Database Extraction Results
Database

Fig. 38

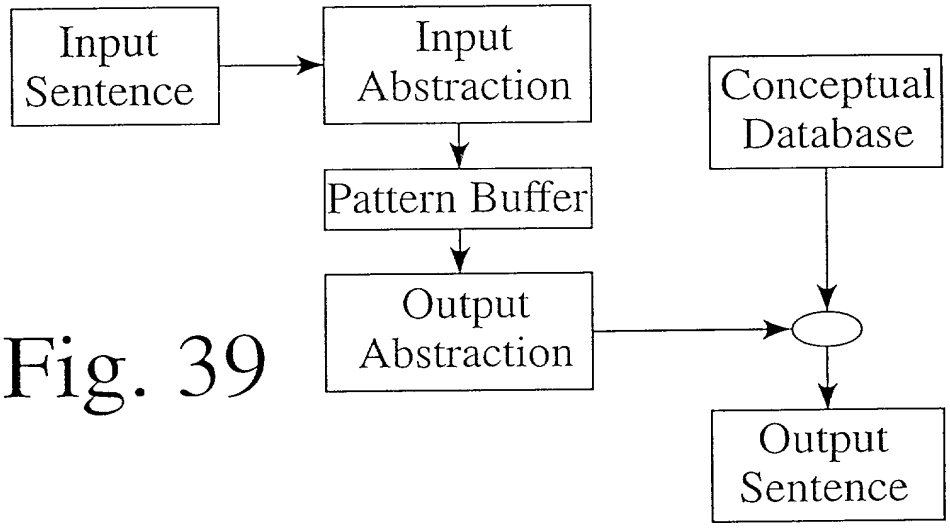

Input Sentence
Input Abstraction
Pattern Buffer
Output Abstraction
Conceptual Database
Output Sentence

Fig. 39

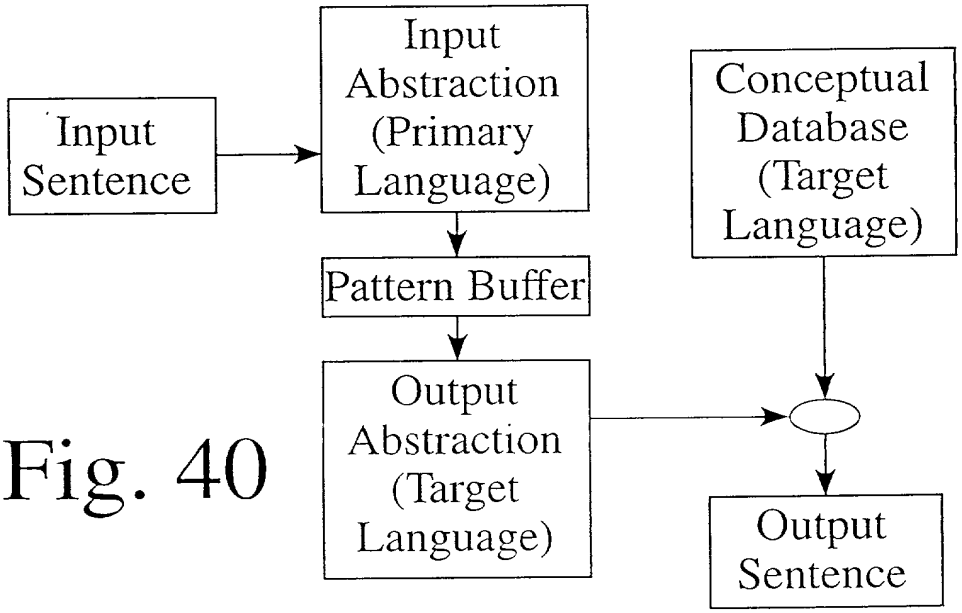
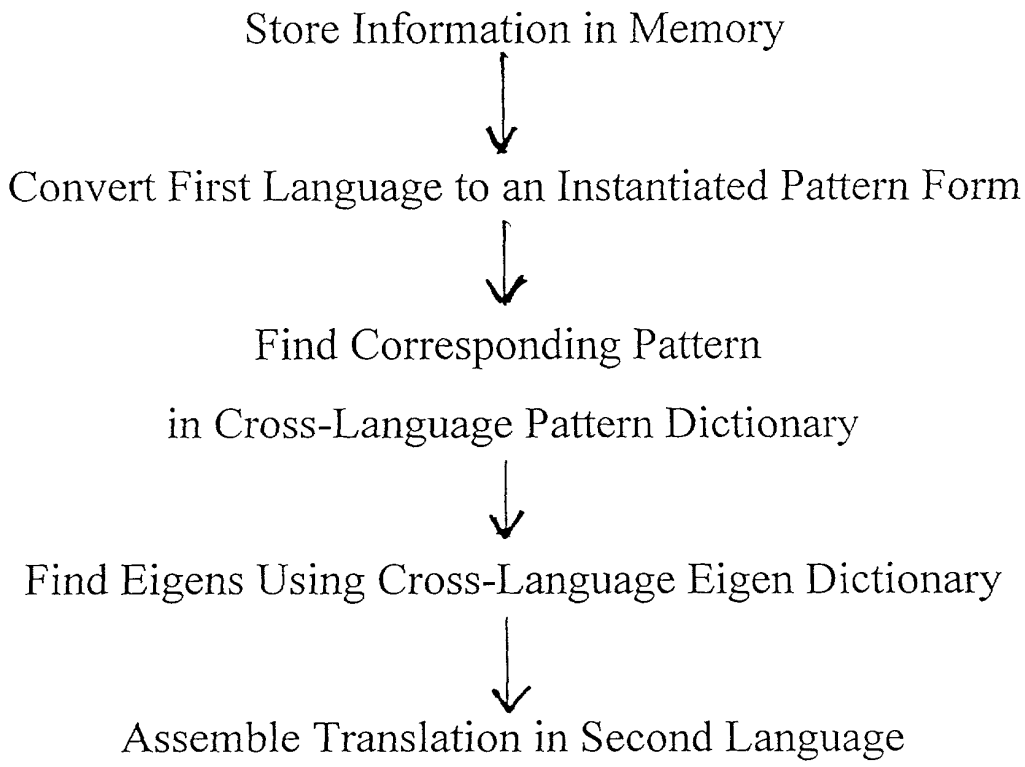

Input Sentence
Input Abstraction (Primary Language)
Pattern Buffer
Output Abstraction (Target Language)
Conceptual Database (Target Language)
Output Sentence

Fig. 40

Fig. 41



Patent Application Publication